UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner

U.S. Patent No. 7,545,740

DECLARATION OF HENRY HOUH,
UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION FOR
***INTER PARTES* REVIEW**

TABLE OF CONTENTS

I. Introduction 5

II. Qualifications and Professional Experience 7

III. Level of Ordinary Skill in the Art 7

IV. Relevant Legal Standards 8

V. Background 10

A. Link Aggregation 10

B. Multiplexing and Demultiplexing 11

VI. The ’740 Patent 16

A. Overview of the ’740 Patent 16

VII. Claim Construction 20

VIII. Identification of how the Claims are Unpatentable 23

A. Ground 1: Claims 1-31 are obvious under 35 U.S.C. § 103(a) over Bruckman in view of Basso. 24

1. Summary of Bruckman 24

2. Summary of Basso 28

3. Reasons to Combine Bruckman and Basso 30

4. Claim 1 36

5. Claim 2 52

6. Claim 3 55

7. Claim 4 57

8. Claim 5 60

9. Claim 6 62

10. Claim 7 64
11. Claim 8 65
12. Claim 9 66
13. Claim 10 67
14. Claim 11 71
15. Claim 12 75
16. Claim 13 77
17. Claim 14 81
18. Claim 15 84
19. Claim 16 85
20. Claim 17 85
21. Claim 18 89
22. Claim 19 90
23. Claim 20 90
24. Claim 21 91
25. Claim 22 92
26. Claim 23 92
27. Claim 24 92
28. Claim 25 92
29. Claim 26 93
30. Claim 27 93
31. Claim 28 93
32. Claim 29 95

33. Claim 30 96

34. Claim 31 97

B. Ground 2: Claims 11 and 26 are obvious under 35 U.S.C. § 103(a) over Bruckman in view of Basso and Holdsworth 97

1. Summary of Holdsworth 34

2. Reasons to Combine Holdsworth with Bruckman and Basso 97

3. Claims 11 and 26 100

IX. Conclusion 101

I, Henry Houh, do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Cisco Systems, Inc. in the matter of the *Inter Partes* Review of U.S. Patent No. 7,545,740 ("the '740 patent") to Zelig *et al*.

2. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this investigation. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether claims 1-31 ("the Challenged Claims") of the '740 patent are unpatentable as they would have been obvious to a person having ordinary skill in the art ("POSITA") at the time of the alleged invention, in light of the prior art. It is my opinion that all of the limitations of the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

a. the '740 patent, Ex.1001;

b. the prosecution history of the '740 patent ("'740 File History"), Ex.1002;

c. U.S. Patent Publication No. 2004/0228278 to Bruckman et al. ("Bruckman"), Ex.1005;

d. U.S. Patent Publication No. 2003/0210688 to Basso et al. ("Basso"), Ex.1006; and

e. Digital Logic Design, by Holdsworth ("Holdsworth") (2002), Ex.1007.

f. In forming the opinions expressed below, I have considered:

U.S. Patent No. 6,943,580 to Lewis et al. ("Lewis"), Ex.1008;

IEEE Standard Dictionary of Electrical and Electronics Terms, (1984), Ex.1009;

RFC 1042, Ex.1010,

C++ Inside & Out, by Bruck Eckel ("Eckel") (1993), Ex.1011.

U.S. Patent No. 7,271,993 to Plojhar ("Plojhar"), Ex.1017;

RFC 793, Transmission Control Protocol, Ex.1018;

RFC 2026, The Internet Standards Process – Revision 3, Ex.1019;

U.S. Patent No. 5,960,428 to Lindsay et al. ("Lindsay"), Ex.1020;

U.S. Patent No. 5,999,538 to Haddock et al. ("Haddock"), Ex.1021; and

A course catalog for electrical engineering, Ex.1022.

the documents listed above;

the relevant legal standards, including the standard for obviousness, and any additional authoritative documents as cited in the body of this declaration; and

my own knowledge and experience based upon my work in the field of optical communication as described below, as well as the following materials.

5. Unless otherwise noted, all emphasis in any quoted material has been added.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

6. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which can be found in Exhibit 1004. The following is a brief summary of my relevant qualifications and professional experience.

III. LEVEL OF ORDINARY SKILL IN THE ART

7. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

8. A person of ordinary skill in the art ("POSITA") in the field of the

’740 patent, as of its earliest possible filing date of April 7, 2006, would have been someone knowledgeable and familiar with the communication network arts that are pertinent to the ’740 patent. A POSITA would have had a bachelor’s degree in electrical engineering, or equivalent training, and approximately two years of experience working in the field of communication networks, including link aggregation in communication networks. Lack of work experience can be remedied by additional education, and vice versa.

9. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my experience and the understanding of a POSITA generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the field as of the priority date of the ’740 patent. Unless otherwise stated, when I provide my understanding and analysis below, it is consistent with the level of a POSITA prior to the priority date of the ’740 patent.

IV. RELEVANT LEGAL STANDARDS

10. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the ’740 patent, I am relying on certain basic legal principles that counsel have explained to me. These principles are discussed below.

11. I understand that prior art to the ’740 patent includes patents and

printed publications in the relevant art that predate the priority date of the alleged invention recited in the ’740 patent. For purposes of this Declaration, I am applying April 7, 2006 as the earliest possible priority date of the ’740 patent.

12. I have been informed that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. I have also been informed by counsel that the obviousness analysis takes into account factual inquiries including the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

13. I have been informed by counsel that the Supreme Court has recognized several rationales for combining references or modifying a reference to show obviousness of claimed subject matter. Some of these rationales include the following: (a) combining prior art elements according to known methods to yield predictable results; (b) simple substitution of one known element for another to obtain predictable results; (c) use of a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a

reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

V. BACKGROUND

A. Link Aggregation

14. As explained by the prior art, "[l]ink aggregation is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link." Bruckman, [0002]. "Link aggregation offers benefits of increased bandwidth, as well as increased availability, since the logical link can continue to function (possibly with reduced bandwidth) even when one of the physical links fails or is taken out of service." Bruckman, [0002]. Link aggregation, for Ethernet networks, is defined by industry standards. *See* Bruckman, [0003] ("[L]ink aggregation is defined by Clause 43 of IEEE Standard 802.3, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (2002 Edition)").

15. When using link aggregation, a mechanism is used to determine which of the physical links will transmit a particular frame. This mechanism includes "a distributor function, which distributes data frames submitted by MAC clients among the physical links in the group, and a collector function, which

receives frames over the aggregated links and passes them to the appropriate MAC clients." Bruckman, [0003]. Industry standards describe "possible distribution algorithms that meet the requirements of the standard, while providing some measure of load balancing among the physical links in the aggregation group." Bruckman, [0005]. These distribution algorithms may include, for example, a hash function. *See* Bruckman, [0006]-[0012].

16. Various hashing techniques for distributing packets or data frames across various link aggregation configurations were thus well-known before the '740 patent was filed.

B. Multiplexing and Demultiplexing

17. Link aggregation uses a concept referred to as multiplexing. A multiplexer selects from a plurality of input lines for transmission over a single output line. As shown in the figure below, a 4:1 multiplexer selects from four possible input lines (d_0, d_1, d_2, d_3) for transmission across a single output line (f). As also shown in the figure below, a 4:1 multiplexer uses a two-bit command signal (denoted A and B) to select from the four possible inputs. A two-bit signal has four possible values (00, 01, 10, 11), each of which corresponds to a different input (d_0, d_1, d_2, d_3).

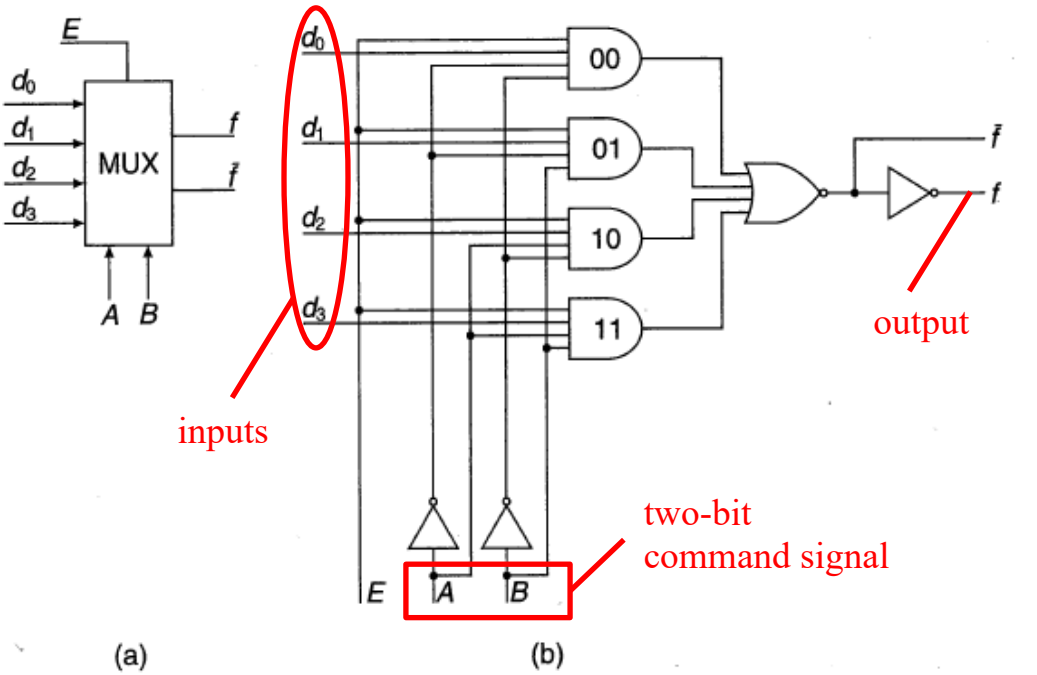


Figure 5.1 *(a) Block diagram of a 4-input multiplexer and (b) its gate implementation*

Holdsworth, Fig. 5.1 (annotated).

18. A demultiplexer is the opposite of a multiplexer—it selects one of a plurality of output lines for transmission of data received over a single input line. As shown in the figure below, a basic 1:4 demultiplexer selects one of four possible output lines (Y1, Y2, Y3, Y4) for a single input line. As also shown in the figure below, a 1:4 demultiplexer uses a two-bit command signal (denoted A and B) to select one of the four possible outputs. Each of the four possible values (00, 01, 10, 11) of the two-bit command signal corresponds to a different output line (Y1, Y2, Y3, Y4).

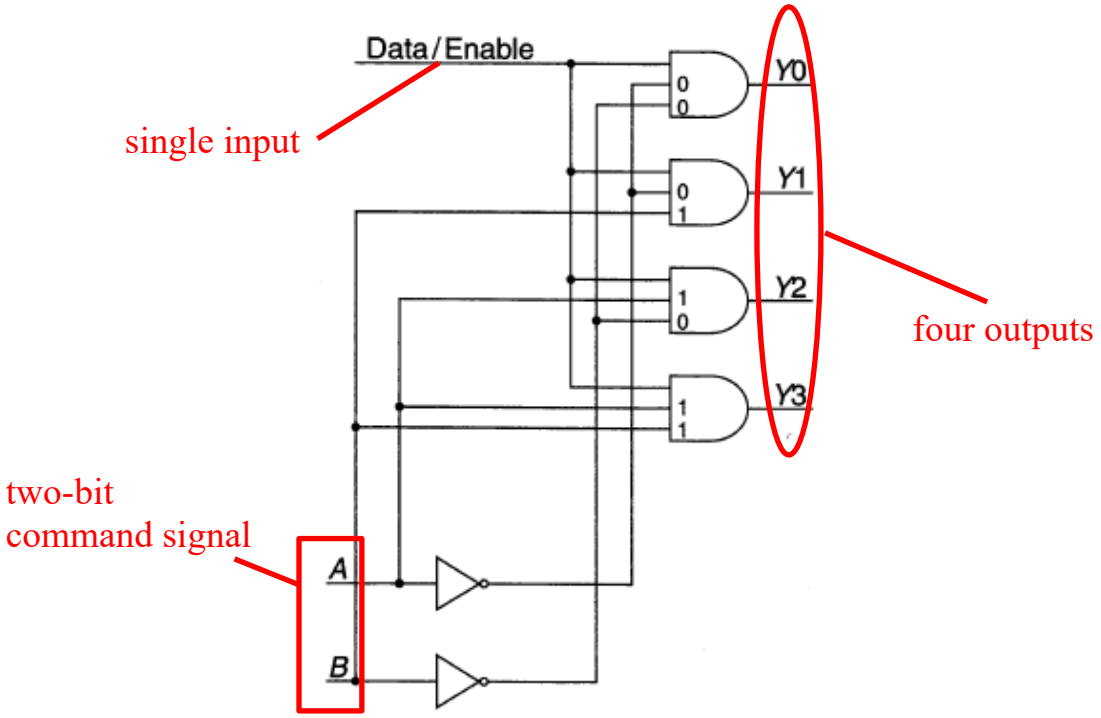
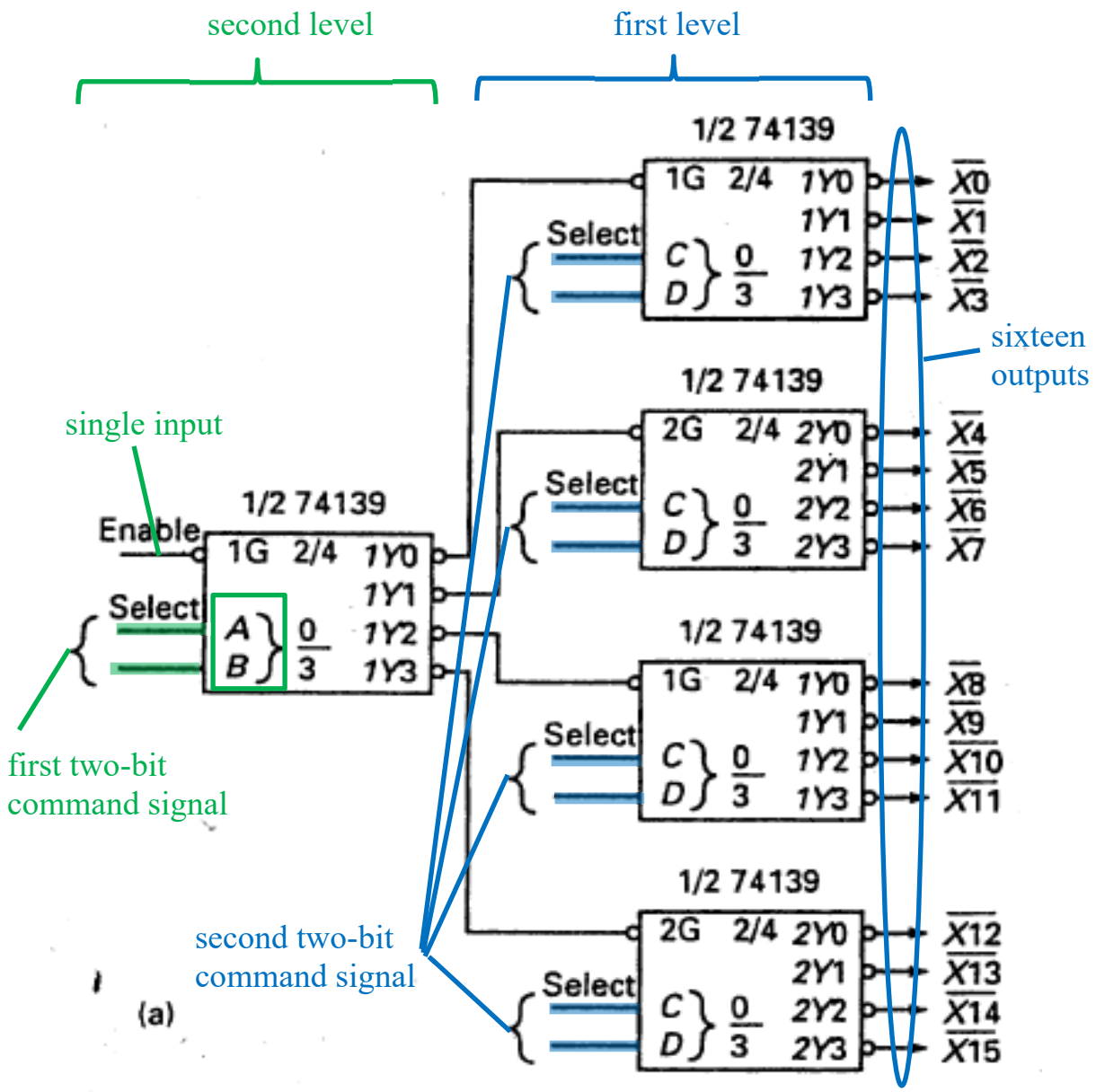


Figure 5.10 *Basic demultiplexer*

Holdsworth, Fig. 5.10 (annotated).

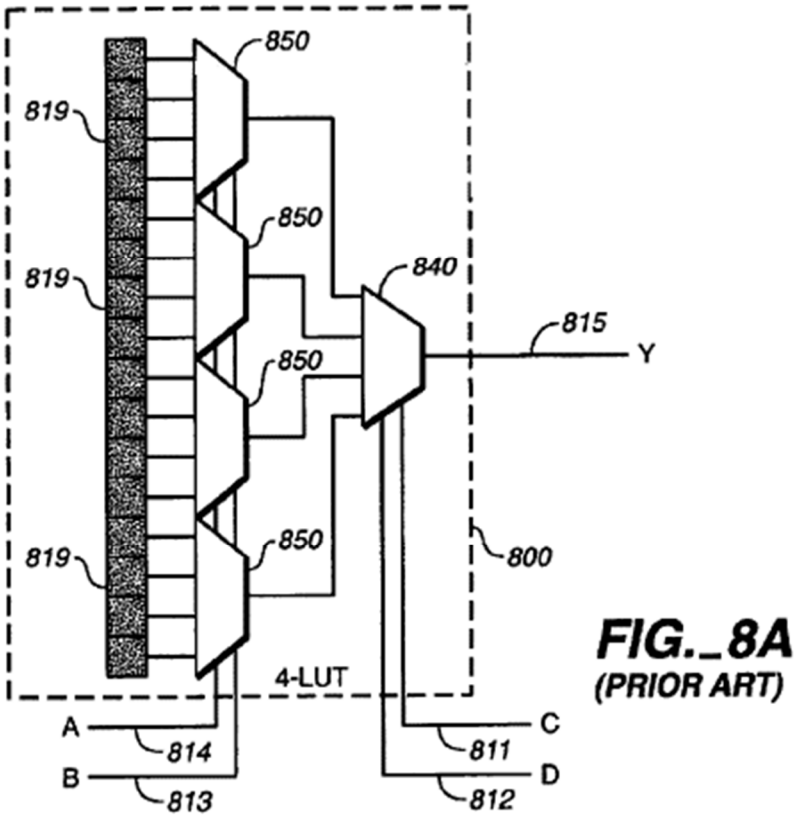
19. To form larger de-multiplexers with more output lines, it was known to use two-level structures comprised of several demultiplexers. For example, as shown in the figure below, it was known to use several 1:4 demultiplexers in a two-level structure to create a 1:16 demultiplexer.



Holdsworth, Fig. 5.17(a) (annotated).

20. As can be seen from the figure below first level (on the right) includes four 1:4 demultiplexers that are controlled by command lines C and D. The second level (on the left) includes a single 1:4 demultiplexer that is controlled by command lines A and B. Through this two-level structure, a four-bit value (A, B, C, D) can select any of the sixteen possible outputs (X0-X15) for a single input. It

was also known to create large multiplexers using a similar two-level structure. *See* Ex.1008, 3:61-7:19; Fig. 8A (illustrating and describing a "conventional" multiplexer).



Lewis, Fig. 8A.

21. Link aggregation uses demultiplexing in that it distributes data traffic from a single input channel among several different links. For example, a 1:16 demultiplexer such as the one shown above can be used to distribute data traffic across 16 different parallel links. When receiving traffic across such links, a multiplexer can be used to combine incoming data from each of the sixteen links to

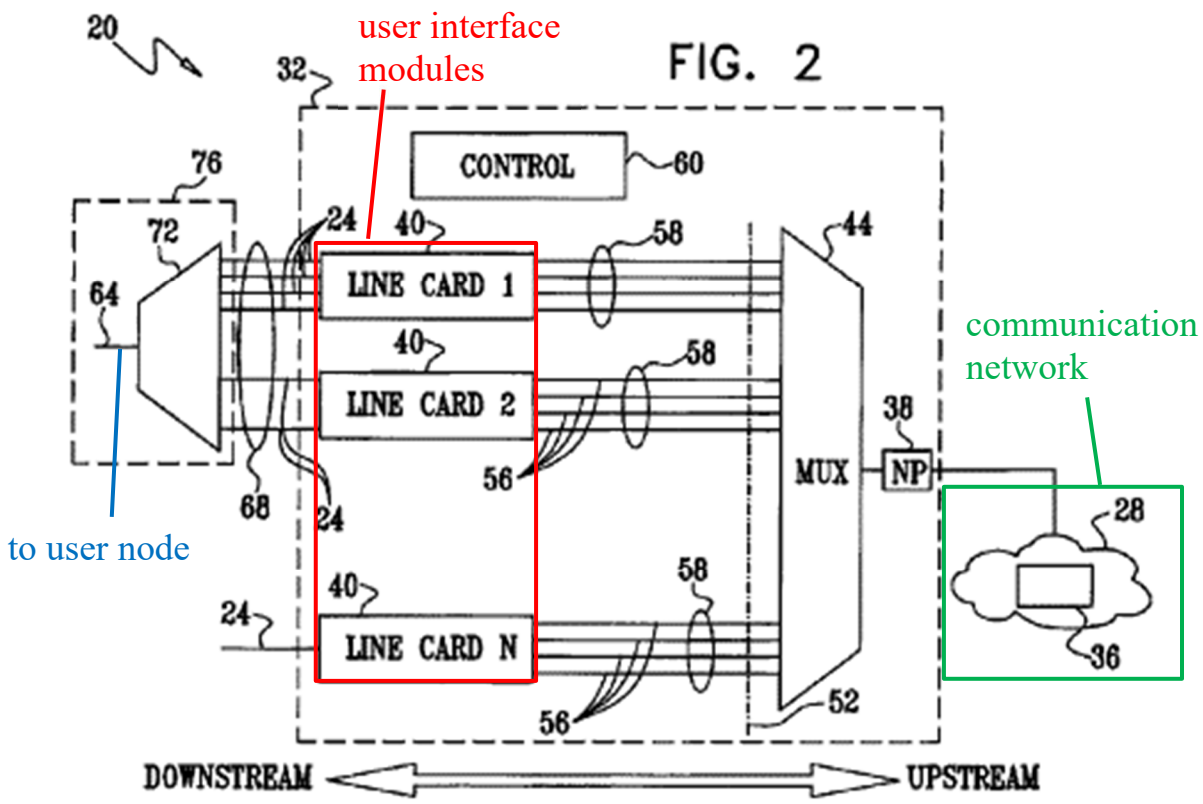
a single channel.

VI. THE '740 PATENT

A. Overview of the '740 Patent

22. The '740 patent is "directed to communication networks, and particularly to methods and systems for link aggregation in network elements." '740 patent, 1:5-7. In the '740 patent, a communication system 20 includes a network element 32 that "interconnects a plurality of user ports 24 to a communication network 28." '740 patent, 4:7-8. "Network 28 may comprise a wide-area network (WAN), such as the Internet, a network internal to a particular organization (Intranet), or any other suitable communication network." '740 patent, 4:9-11. The user ports may connect "to a user node, such as a layer 2 or layer 3 switch." '740 patent, 6:38-39.

23. The network element 32 includes "one or more user interface modules (UIMs), such as line cards 40. Each line card is assigned to process data frames of one or more user ports." '740 patent, 4:28-30. Network element 32 is shown below in Fig. 2.

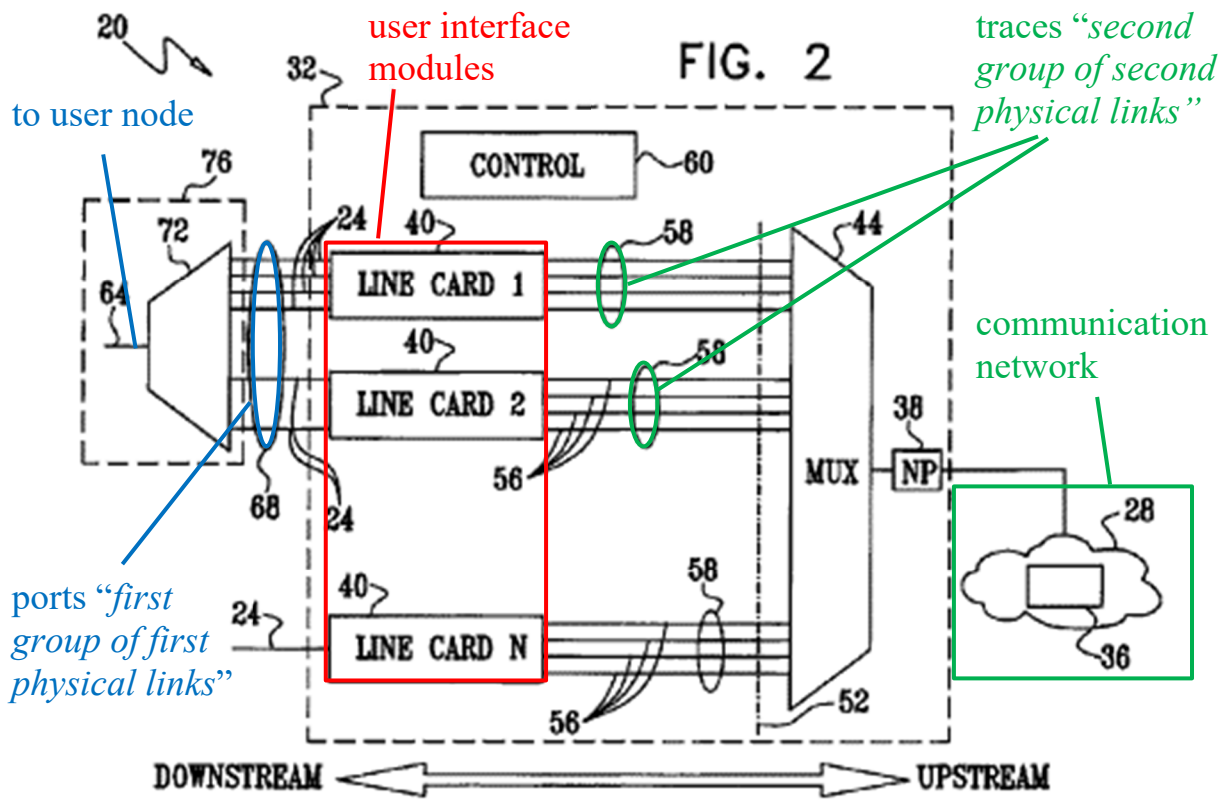


'740 patent, Fig. 2 (annotated).

24. The line cards 40 are connected to the user node through a set of ports 24. *See* '740 patent, 4:5-11, Fig. 2. "User ports 24 forming port 64 are configured as an Ethernet LAG [link aggregation group] group, referred to as an external LAG group 68." '740 patent, 6:21-23. The '740 patent claims these ports 24 as "*a first group of first physical links arranged in parallel*." *See e.g.,* '740 patent, claim 1.

25. The interface modules are connected to the communication network 28 using backplane traces 56: "Backplane 52 comprises physical links, such as backplane traces 56, typically in the form of printed circuit board (PCB) conductors. Each backplane trace 56 has a finite bandwidth and can support a

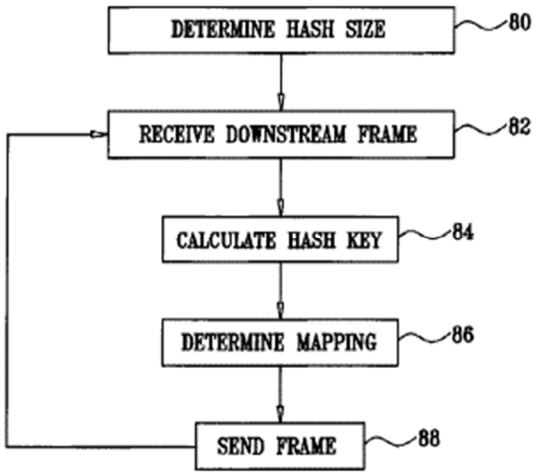
certain maximum frame throughput." '740 patent, 4:34-37. The '740 patent claims these traces 56 as "*a second group of second physical links arranged in parallel*." *See e.g.,* '740 patent, claim 1.



'740 patent, Fig. 2 (annotated).

26. The network element 32 further includes a control module 60 that, for each received packet, determines which trace and which port will transmit that packet. The '740 patent explains that selection of both a port 24 from the first group and a trace 56 from the second group can be done in "single combined mapping operation that combines the two mapping operations described above." '740 patent, 6:63-65. The "combined mapping comprises a single hashing

operation that determines, for each such downstream frame, both the backplane trace 56 over which the frame is to be sent to one of line cards 40, and the user port 24 to be used within external LAG group 68." '740 patent, 6:66-7:3. This combined mapping operation includes several steps, as shown below in Fig. 4.



'740 patent, Fig. 4.

27. At step 80, "[t]he method begins with control module 60 determining a hashing size parameter." '740 patent, 7:25-26. The hashing size parameter is based on the total number of traces multiplied by the total number of ports. *See* '740 patent, 7:42. Then, at step 82, a data frame is received. '740 patent, 7:43-45. "For each downstream frame, control module 60 calculates a hashing key of the frame, at a hash key calculation step 84. As explained above, the hashing key is typically calculated by applying a suitable hashing function to the frame attributes of the downstream frame." '740 patent, 7:45-50. At step 86, "Module 60 divides the hashing key of the downstream frame by N_{bpow} [the hashing size parameter]

and retains the modulo, or the remainder of the division operation, as a mapping index." '740 patent, 7:51-53.

28. Accordingly, the steps shown in Fig. 4 are used to produce a single hash computation that is then used to determine both a port and a trace to transmit the packet. The '740 patent provides an example in which "Module 60 partitions the binary representation of the mapping index into two parts having N1 and N2 bits." '740 patent, 7:60-61. The first set of bits N1 is used to select one of the ports: "Module 60 uses N1 bits as a user slot/port index, indicating over which user port 24 in external LAG group 68 the frame should be sent." '740 patent, 7:61-64. The second set of bits N2 is used to select one of the traces: "The remaining N2 bits are used as a backplane trace index, indicating over which of the backplane traces of the relevant line card the frame should be sent." '740 patent, 7:64-66.

29. The concept of a network element that provides link aggregated ports to connect a user node to a network such as the Internet was already known at the time the '740 patent was filed. It was also known at the time to select both a port and a trace using a single hash computation, as will be described in further detail below.

VII. CLAIM CONSTRUCTION

30. It is my understanding that in order to properly evaluate the '740 patent, the terms of the claims must first be interpreted. It is my understanding that

for the purposes of this *inter partes* review, the claims are to be construed under the so-called *Phillips* standard, under which claim terms are given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry.

31. I have reviewed the entirety of the '740 patent, as well as its prosecution history. It is my opinion that for purposes of applying the prior art presented herein to evaluate the patentability the claim terms do not require express construction.

32. Below are specific terms that I assessed for clarity. I have applied the so-called *Phillips* standard, which requires that claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term.

A. "*interface module*"

33. Each independent claim of the '740 patent recites "*one or more interface modules*." The '740 patent explicitly contemplates that a line card is within the scope of the term interface module: "Network element 32 comprises one or more user interface modules (UIMs), such as line cards 40." '740 patent, 4:28-

29. A line card is thus an interface module as claimed.

B. "*selecting, in a single computation*"

34. As explained above at VI.A, the '740 patent describes a multi-step selecting process as shown in Fig. 4 and explained in accompanying text. This multi-step process includes determining a hash size (step 80), calculating a hash key (step 84), and determining a mapping in a single computation (step 86). This multi-step process is recited in various dependent claims—including claims 8-10 which recite that the selecting process includes multiple steps, including (1) "determining a hashing size responsively to a number of at least some of the first and second physical links," (2) "applying the hashing function to the at least one of the frame attributes to produce a hashing key," and (3) "calculating a modulo of a division operation of the hashing key by the hashing size."

35. Because the claimed selecting process involves multiple steps, as claimed and as described in the specification, a POSITA would have understood the term "*selecting,* ***in a single computation***" one of two ways. In one interpretation, a POSITA would have understood that the claimed "*single computation*" refers to using a single computational result (i.e., a single value) and does *not* refer to performing the selecting process in a single step. Alternatively, a POSITA would have understood the claimed "*single computation*" to refer to only a specific step within the multi-step selecting process—the modulo function

calculation step described in step 86.

36. As will be explained in detail below, a substantively identical selecting process for a two-level demultiplexer structure as described and claimed in the ’740 patent is described in the prior art. Thus, regardless of how a “*single computation*” is interpreted, it is rendered obvious by the prior art.

VIII. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

37. I have been asked to provide my opinion as to whether the Challenged Claims of the ’740 Patent would have been obvious in view of the prior art. The discussion below provides a detailed analysis of how the prior art references identified below teach the limitations of the Challenged Claims of the ’740 patent. As part of my analysis, I have considered the scope and content of the prior art and any differences between the alleged invention and the prior art. I describe in detail below the scope and content of the prior art, as well as any differences between the alleged invention and the prior art, on an element-by-element basis for each Challenged Claims of the ’740 patent.

38. As described in detail below, the alleged invention of the Challenged Claims would have been obvious in view of the teachings of the identified prior art references as well as the knowledge of a POSITA. I rely on various documents to show the background knowledge of a POSITA. Such documents include RFCs, which are “Request for Comments” (RFC) documents published by the Internet

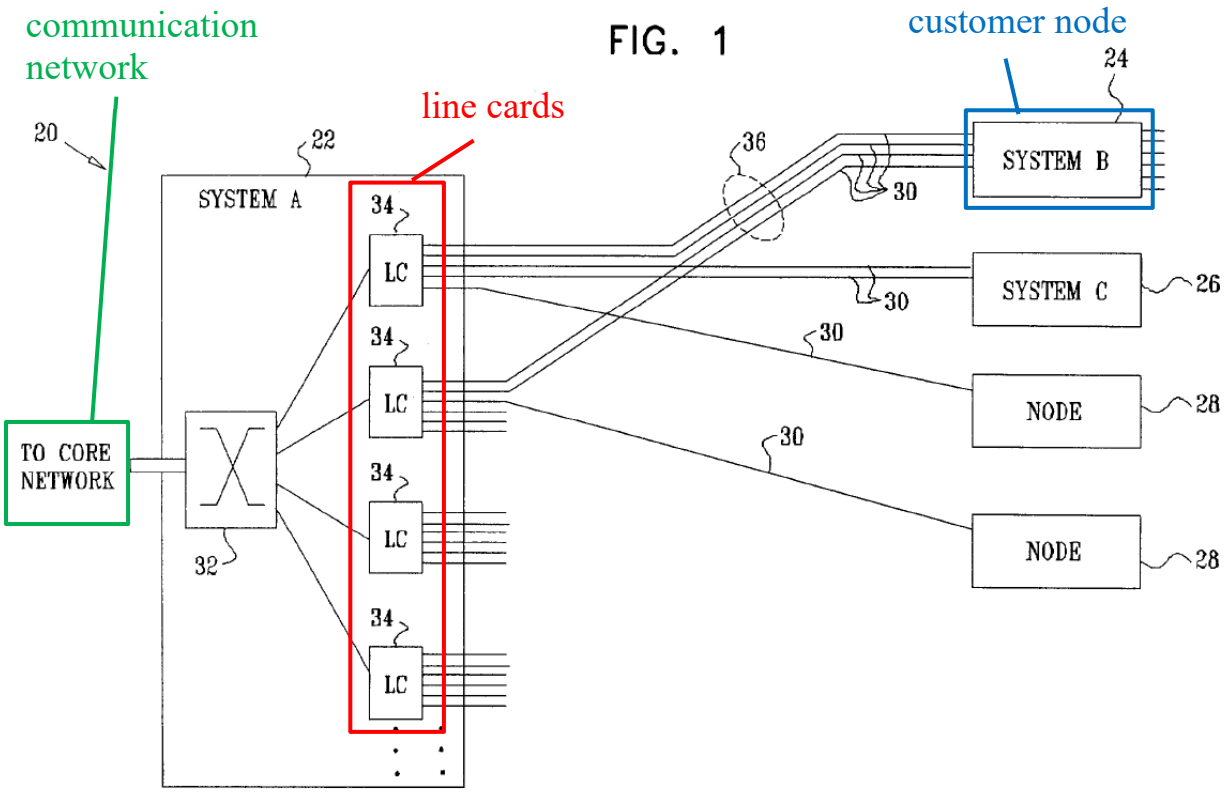
Engineering Task Force (IETF). RFCs can be obtained from a number of Internet hosts. Ex.1019, 6. IETF documents are "readily available to a wide audience." Ex.1019, 8. IETF participants "shall publicly announce…every activity" relating to the standardization process. Ex.1019, 26.

39. Analysis for Grounds 1 and 2 is very similar, and thus they are presented together. Ground 2 differs from Ground 1 by relying on additional obviousness teachings from Basso for limitation [1.6].

A. Grounds 1 & 2: Claims 1-31 are obvious under 35 U.S.C. § 103(a) over Bruckman, alone or in view of Basso.

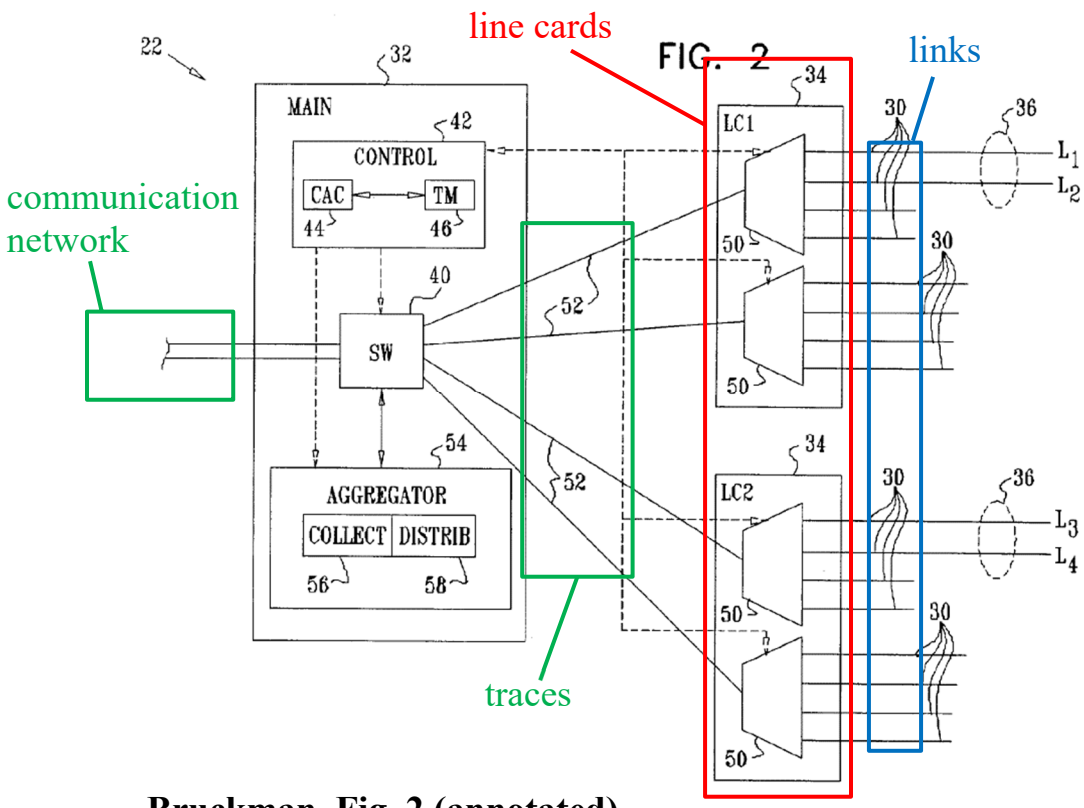
1. Summary of Bruckman

40. Like the '740 patent, Bruckman relates "to data communication systems, and specifically to methods and systems for link aggregation in a data communication network." Bruckman, [0001]. Bruckman describes a communication system 20 having Equipment 22 that "is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example)." Bruckman, [0047]. Equipment 22 further includes "a main switching card 32, which is connected to multiple line cards 34." Bruckman, [0047]. As shown in Fig. 1 below, the main switching card 32 and line cards 34 form a two-level multiplexer/demultiplexer.



Bruckman, Fig. 1 (annotated).

41. As shown in Fig. 2, which provides more details of the communication system 20 shown in Fig. 1, the line cards connect to the customer node via links 30. "Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30." Bruckman, [0056]. The line cards 34 connect to the switching core 40 and core network through backplane traces 52. Bruckman thus describes a two-level multiplexing/ demultiplexing structure in which the first level includes the concentrators 50, each of which selects from one of four outputs (links 30) and the second level includes switching core 40, which selects one of a plurality of outputs (traces 52).



Bruckman, Fig. 2 (annotated).

42. When processing traffic from the customer nodes to the communication network, Bruckman's two-level structure acts as a multiplexer: "The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40." Bruckman, [0056]. When processing traffic from the communication network to a customer node, Bruckman's two-level structure acts as a demultiplexer by determining which trace and which port will transmit that data frame. The main card 32 includes a distributor 58, that "determines the link over which to send each frame based on information in the frame header." Bruckman, [0058]. To make this selection, the

distributor "applies a predetermined hash function to the header information." Bruckman, [0058]. Bruckman provides an example of such a hash function shown in Table 1 below.

```
A  unsigned short hash(unsigned char *hdr, short lagSize)
       {
       short i;
       unsigned short hash=178; // initialization value
       for (i=0; i<4; i++)          // hdr is 4 bytes length
B         hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
C      return (hash % lagSize)
       }
```

Bruckman, Table 1 (annotated).

43. Like the selection process in the '740 patent, Bruckman's hashing function includes multiple steps that produce a single hash computation. First, as shown in line A, the hashing function receives the lagSize parameter, which "is the number of active ports (available links 30) in link aggregation group." Bruckman, [0064]. In line B, an operation is applied to header information (in variable *hdr[1]) to create a hashing key stored in the "hash" variable. Then, in line C, a modulo of the division of the hash key and the LAG size is produced. This final, single

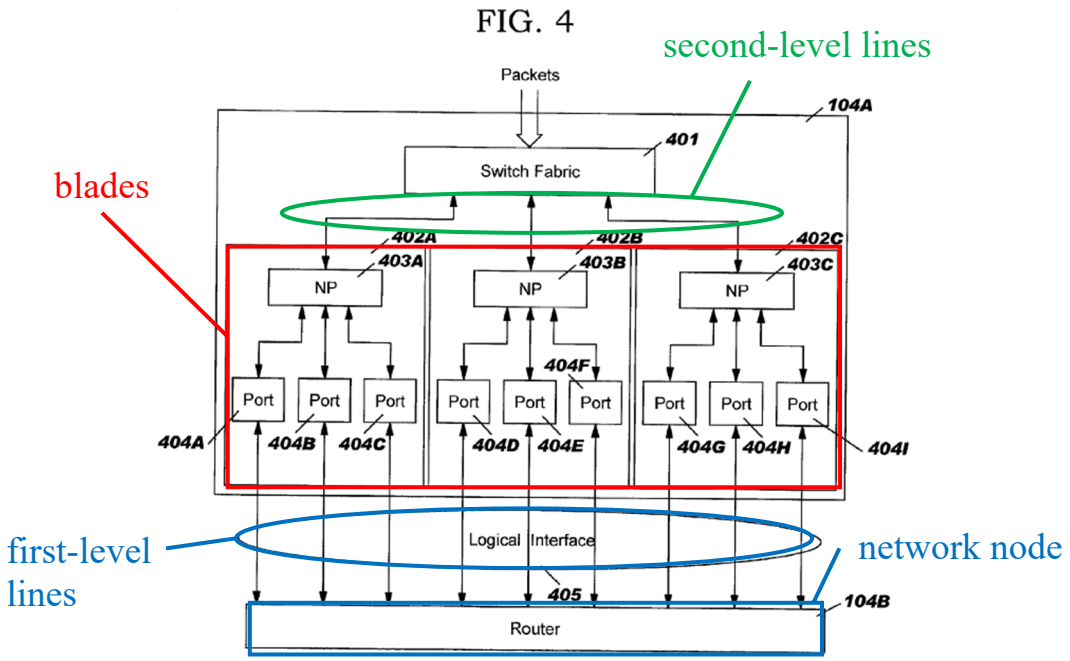
[1] "Here hdr is the header of the frame to be distributed." Bruckman, [0064].

modulo value is then used to select which link 30 will transmit the data frame.

44. Accordingly, Bruckman shows that using a set of parallel traces and a set of parallel links/ports to transmit packets was known before the filing of the ’740 patent. Bruckman additionally shows that it was known to apply a hashing function to select one of the links.

2. Summary of Basso

45. Like both Bruckman and the ’740 patent, Basso relates to link aggregation, particularly, “the field of packet switching networks, and more particularly to logically grouping physical ports of a network device, e.g., router, into logical interfaces thereby expanding the bandwidth capacity of transmitting packets from that network device to another network device.” Basso, [0001]. Basso describes a network device, such as a router, that passes packets between the Internet and a client device. Like Bruckman, Basso describes a two-level structure, as shown in Fig. 1 below.



Basso, Fig. 4 (annotated).

46. Basso's network device 104a includes "a switch fabric coupled to a plurality of blades where **<u>each blade may comprise</u>** one or more network processors coupled to **<u>one or more ports</u>**." Basso, [0009]. As shown in Fig. 4, Basso's two-level structure includes a first group of physical links—each associated with one of the ports 404—that connect the network device to a network node (in this case, Router 104b). Additionally, a second set of links connects each of the blades to the switching fabric: "[A] switch fabric 401 configured to direct the incoming packets of data to particular **<u>blades 402A-C coupled to switch fabric 401</u>**." Basso, [0030].

47. Upon receipt of a packet, "a hash function may be performed on the source and destination address in the packet header to generate a hash value."

Using that single hash value, Basso's network device selects "an appropriate **blade/port combination** may be identified to transmit the received packet of data." Basso, [0012]. Basso thus uses a single hash computation to select both a first-level line (one of the ports) and second-level line (one of the blades)—a blade/port combination.

3. Reasons to Combine Bruckman and Basso

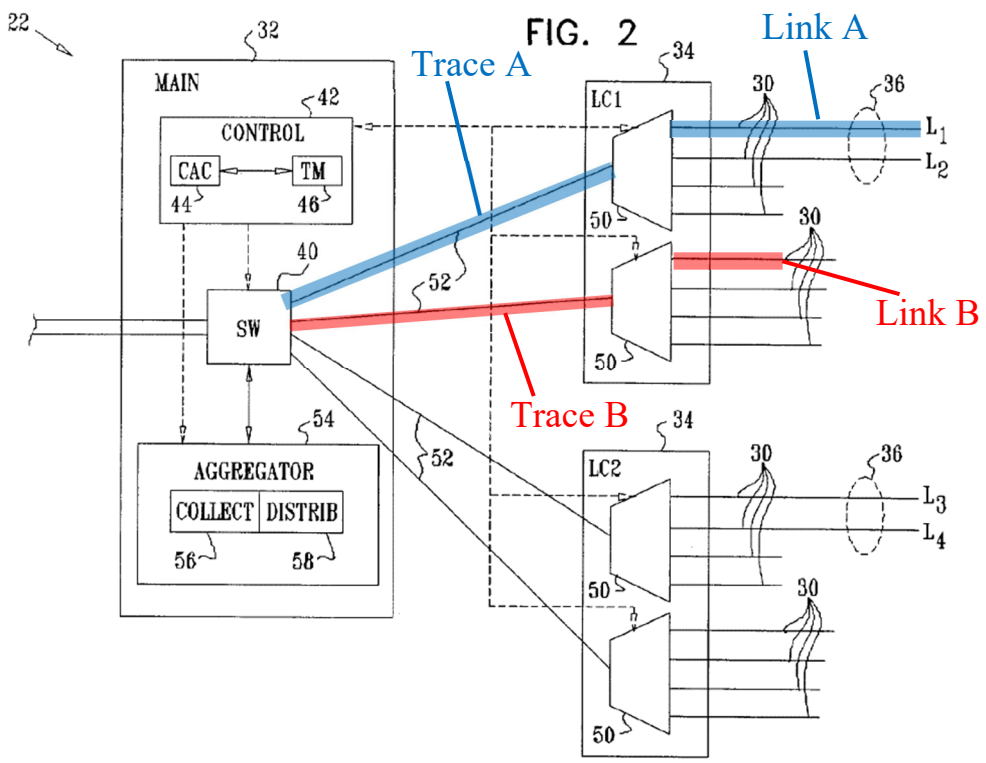
48. A POSITA would have found it obvious that when using a two-level demultiplexer structure such as the one described by Bruckman, a single hash computation would be used to control *both* a first-level and a second-level demultiplexer. Basso provides explicit evidence of doing so. Both Bruckman and Basso are directed to the same field of endeavor as the '740 patent—network communications. *Compare* Ex.1001, 1:5-7 ("The present invention relates generally to communication networks, and particularly to methods and systems for link aggregation in network elements") *with* Ex.1005, [0001] ("The present invention relates generally to data communication systems, and specifically to methods and systems for link aggregation in a data communication network.") *and* Ex.1006, [0001] ("The present invention relates to the field of packet switching networks, and more particularly to logically grouping physical ports of a network device, e.g., router, into logical interfaces").

49. Bruckman's hashing technique produces a single hash computation

that is used determine which link 30 receives a data frame: The "distributor 58, which is responsible for distributing data frames arriving from the network **among links 30** in aggregation group 36 … determines **the link** over which to send each frame based on information in the frame header." Bruckman, [0058].

50. Bruckman provides limited description of how the hash computation is used to select a trace. A POSITA would have recognized, however, that Bruckman's selection of a link 30 also determines which trace 52 will transmit the data frame because each link 30 is associated with only a single trace 52. Referring to Fig. 2 below, if Link A is selected by the hash function, then Trace A is also selected since Trace A is the only trace that connects to link A to the switching core.[2] Similarly, if Link B is selected by the hash function, then Trace B is also selected, since Trace B is the only trace that connects link B to the switching core. Accordingly, selection of a specific link 30 also determines a specific trace 52.

[2] Because Bruckman identifies all links or all traces with like reference numerals, I have labeled and refer to specific traces and links as Trace A, Trace B, Link A, and Link B.



Bruckman, Fig. 2 (annotated).

51. Basso provides further details regarding such two-level selections and includes explicit evidence of using a single hash computation to make a selection in both levels of a two-level multiplexer/demultiplexer structure. As explained above, Basso's network device uses one hash value to select "an appropriate **<u>blade/port combination</u>** … to transmit the received packet of data." Basso, [0012]. Basso thus uses a single hash computation to select a blade/port combination—both a first-level line (one of the ports) and second-level line (one of the blades). Accordingly, to the extent a POSITA implementing the device of Bruckman needed more information about how a single hash computation would

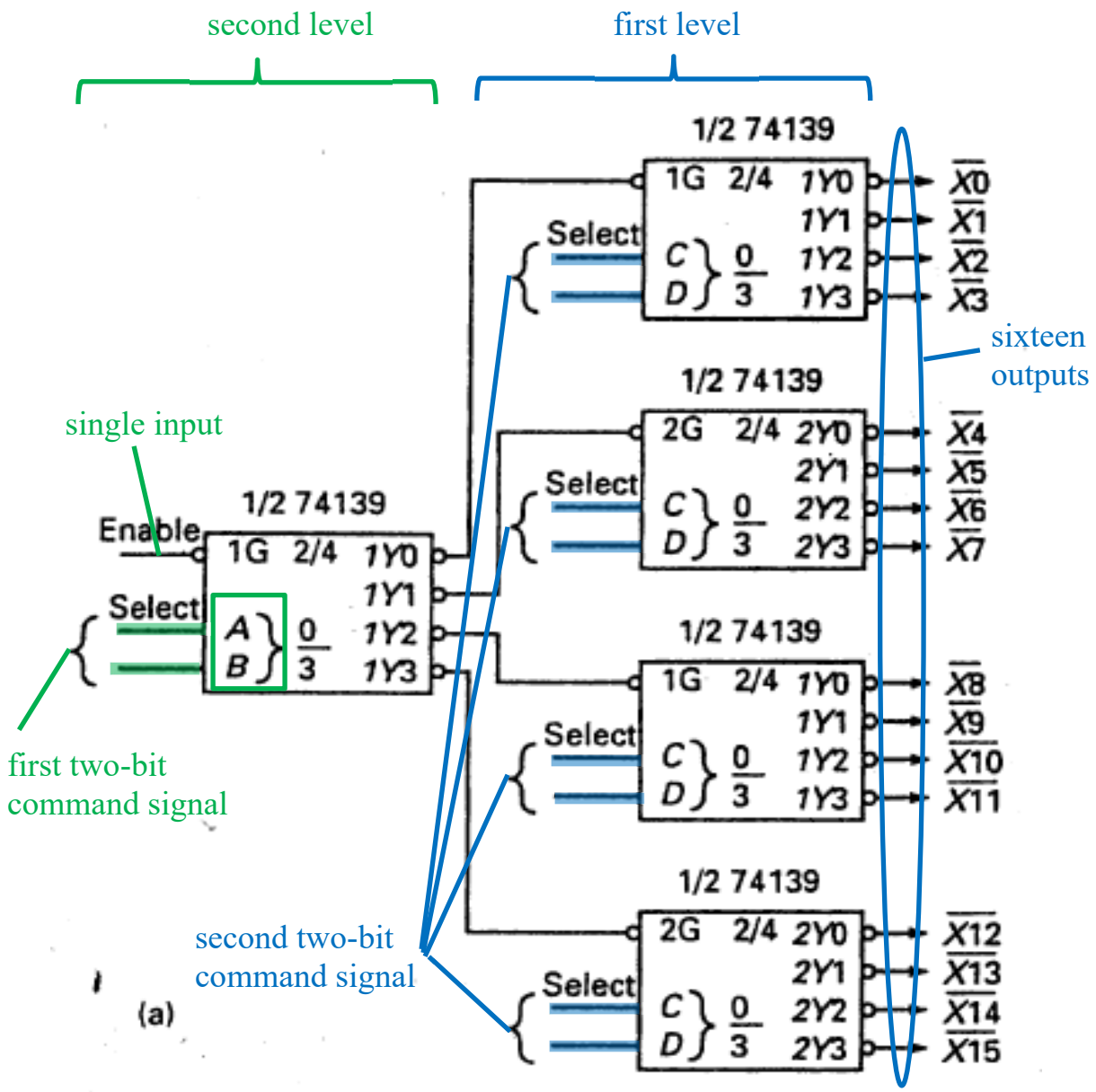
be used to control both levels of a two-level multiplexer/demultiplexer structure, such teaching is provided by Basso.

52. A POSITA would have had a reasonable expectation of success because Basso describes a substantially similar two-level multiplexer/demultiplexer structure as Bruckman. Thus, using a single hash computation to make selections in both levels of the two-level multiplexer/demultiplexer structure would have worked as well in Bruckman. A POSITA would have further found the combination beneficial as using a single hash computation (rather than multiple) would have allowed for improved operational efficiency due to the simplified design. Calculating a single hash value as opposed to multiple hash values reduces the computational load and thus allows for faster switching, which is consistent with Bruckman's goal to "ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees." Bruckman, [0015].

53. Thus, the combination of Bruckman and Basso represents the application of a known technique (Basso's single hash computation to select both levels in a two-level multiplexer/demultiplexer structure) to Bruckman's method (using a single hash computation for a two-level multiplexer/demultiplexer structure) to yield predictable results (fast switching).

4. Summary of Holdsworth

54. Holdsworth is a textbook entitled "Digital Logic Design." The Holdsworth textbook is "intended to cover all the material that is needed in a typical undergraduate or Master's course on Digital Logic Systems, and also to act as a reference text for graduates working in this field." Holdsworth, preface. Digital Logic Systems is a commonly required course for an undergraduate degree in electrical engineering. *See e.g.,* Ex.1022. As explained above at V.B, Holdsworth describes a base multiplexer and a basic demultiplexer. *See* Fig. 5.1 and 5.10. Holdsworth further describes how 1:4 demultiplexers may be used in a two-level fashion to create a 1:16 multiplexer, as shown in Fig. 5.17(a) below.



Holdsworth, Fig. 5.17(a) (annotated).

55. Holdsworth thus shows that when using two levels of 1:4 demultiplexers to create a 1:16 demultiplexer, a designer would use a two-bit signal for selection at the first level and a second two-bit signal for selection at the second level. In other words, Holdsworth shows that it was a well-known technique to use a first subset of bits for selection at the first level of a two-level

demultiplexer, and a second subset of bits for selection at the second level of the two-level demultiplexer.

56. Holdsworth's engineering principles are applicable to the same field of endeavor as the '740 patent—communication networks. As explained above in my background section, link aggregation in communication networks utilizes multiplexers, which Holdsworth describes in detail. *See generally,* Ex.1007. Because Holdsworth provides implementation details for multiplexers and demultiplexers, Holdsworth is reasonably pertinent to the technical problem of load balancing and link aggregation to which the '740 patent relates.

5. Holdsworth Shows the Background Knowledge of a POSITA

57. The Holdsworth textbook is "intended to cover all the material that is needed in a typical undergraduate or master's course on Digital Logic Systems, and also to act as a reference text for graduates working in this field." Ex.1007, preface. Holdsworth represents the background knowledge of a POSITA because Holdsworth's subject matter would have been part of the undergraduate education of a POSITA. For example, digital logic systems was a commonly required course in undergraduate electrical engineering degree plans. *See* Ex.1022.

6. Claim 1

[1.0] ***A method for communication, comprising:***

58. Bruckman "relates generally to data communication systems, and

specifically to **methods** and systems for link aggregation in a **data communication** network." Bruckman, [0001].

> A method for establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints. A link bandwidth is allocated on each of the physical communication links so as to include a predefined safety margin, based on either a failure protection policy, or a measure of fluctuation that occurs in a rate of data transmission over the physical links, or both. A sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection. The data are conveyed over the logical link by distributing the data for transmission among the physical links in accordance with the allocated link bandwidth.

Bruckman, abstract.

59. Thus, because Bruckman describes methods for link aggregation in a data communication network, Bruckman renders obvious a "*method for communication*" as claimed.

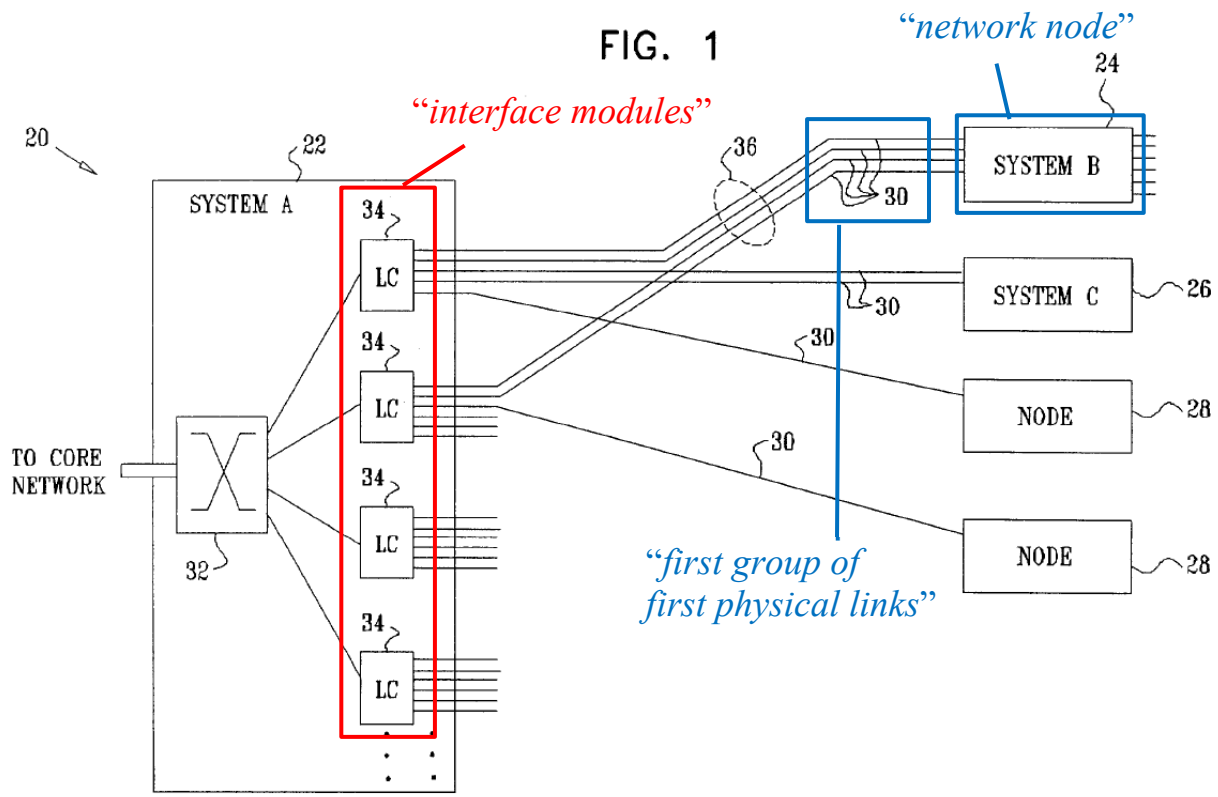
[1.1] *coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,*

60. Bruckman describes a communication system that includes a set of customer nodes 24, 26, 28 (any of which is a "*network node*") that are connected to multiple line cards 34 ("*one or more interface modules*") over parallel physical

links 30 ("*a first group of physical links arranged in parallel*").

> FIG. 1 is a block diagram that schematically illustrates elements of a communication system 20, in accordance with an embodiment of the present invention. **In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, . . . , over physical links 30**. Links 30 typically comprise full-duplex Ethernet links, such as 10BASE-n, 100BASE-n or Gigabit Ethernet links, as are known in the art. (Alternatively, as noted above, other types of physical links may be used, such as ATM or PPP links.) Equipment 22 is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example). **For this purpose, equipment 22 comprises a main switching card 32, which is connected to multiple line cards 34 that serve links 30**. Details of the structure and operation of equipment 22 are shown below in FIG. 2 and are described with reference thereto.

Bruckman, [0047].



Bruckman, Fig. 1 (annotated).

61. As shown in the figure, the links 30 may form an "aggregation group 36." Bruckman, [0057]. Bruckman explains that "Link aggregation is a technique by which a group of **<u>parallel physical links</u>** between two endpoints in a data network can be joined together into a single logical link." Bruckman, [0002].

62. Thus, because Bruckman describes connecting a customer node to a set of line cards over a set of parallel links, Bruckman renders obvious "*coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel*" as claimed.

[1.2] *at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;*

63. Bruckman's Equipment 22 transmits data over the physical links 30 in both directions. Bruckman states that the links are full-duplex links:

> In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, ..., over physical links 30. **Links 30 typically comprise full-duplex Ethernet links**, such as 10BASE-n, 100BASE-n or Gigabit Ethernet links, as are known in the art.

Bruckman, [0047].

64. The term "full-duplex," in the context of an Ethernet link, refers to a link that is capable of simultaneously transmitting data in both directions. *See* IEEE Dictionary, 376 ("full duplex (data transmission) … Refers to a communication system or equipment capable of transmission simultaneously in two directions"); *see also* Ex.1021, 8:11-13 ("full-duplex operation, i.e., carrying data in both directions (to and from the node) simultaneously without corrupting data"). Accordingly, the links 30 transmit data in both directions.

65. For data traffic in the downstream direction (from communication network to customer node), Bruckman's equipment 22 uses a distributor 58: "Aggregator 54 comprises **a distributor 58, which is responsible for distributing data frames arriving from the network among links 30** in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header." Bruckman, [0058]. In the upstream direction (from customer node to communication network), Bruckman utilizes a collector

56: "Aggregator 54 further comprises a collector 56, **which collects data frames that were received over different links 30** in group 36, and arranges the frames back into a single traffic stream." Bruckman [0065].

66. Thus, because Bruckman describes traffic flowing in both directions across the links, Bruckman renders obvious "*at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction*" as claimed.

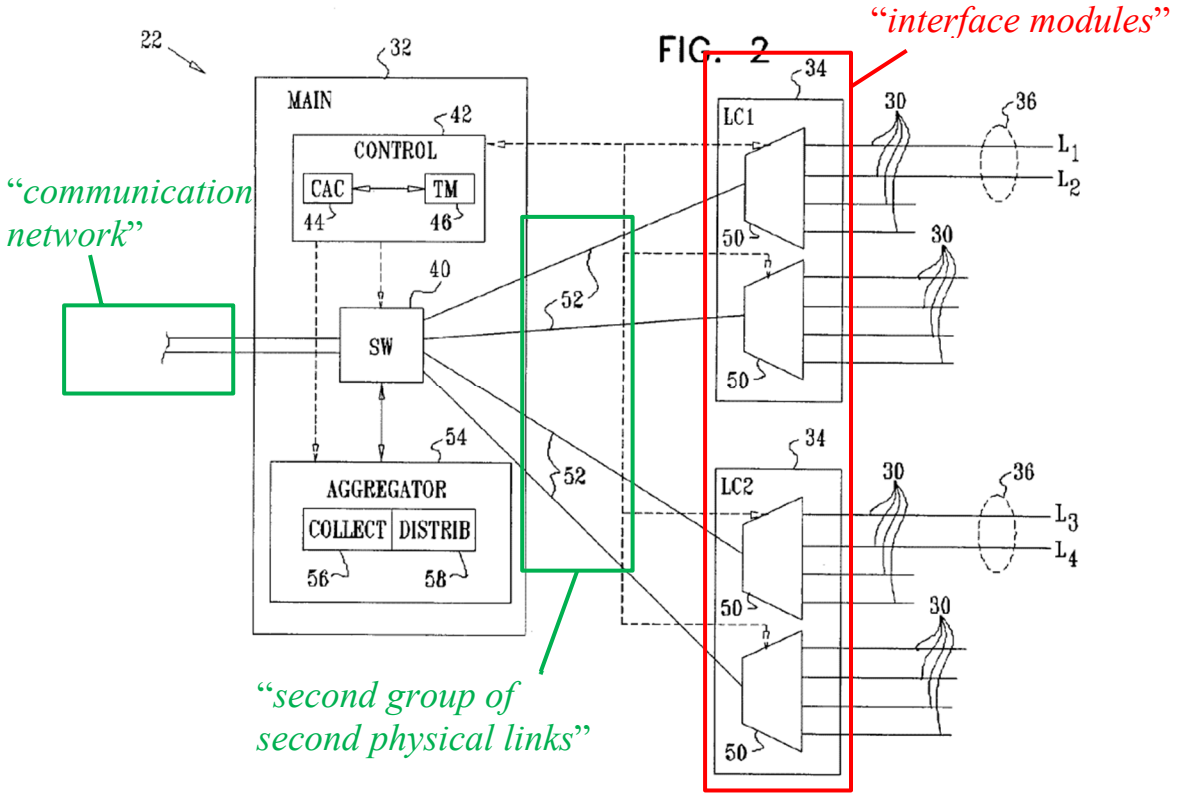
[1.3] *coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel*,

67. In Bruckman's system, the line cards 34 ("*one or more interface modules*") are coupled to the core network ("*communication network*") with a set of parallel traces 52 ("*a second group of second physical links arranged in parallel*").

> Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30. **The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40**. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of statistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of

> incoming data admitted on each link 30 so that it remains between a predetermined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the operational status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concentrators 50, based on the status information and the bandwidth allocations made by CAC 44.

Bruckman, [0056].

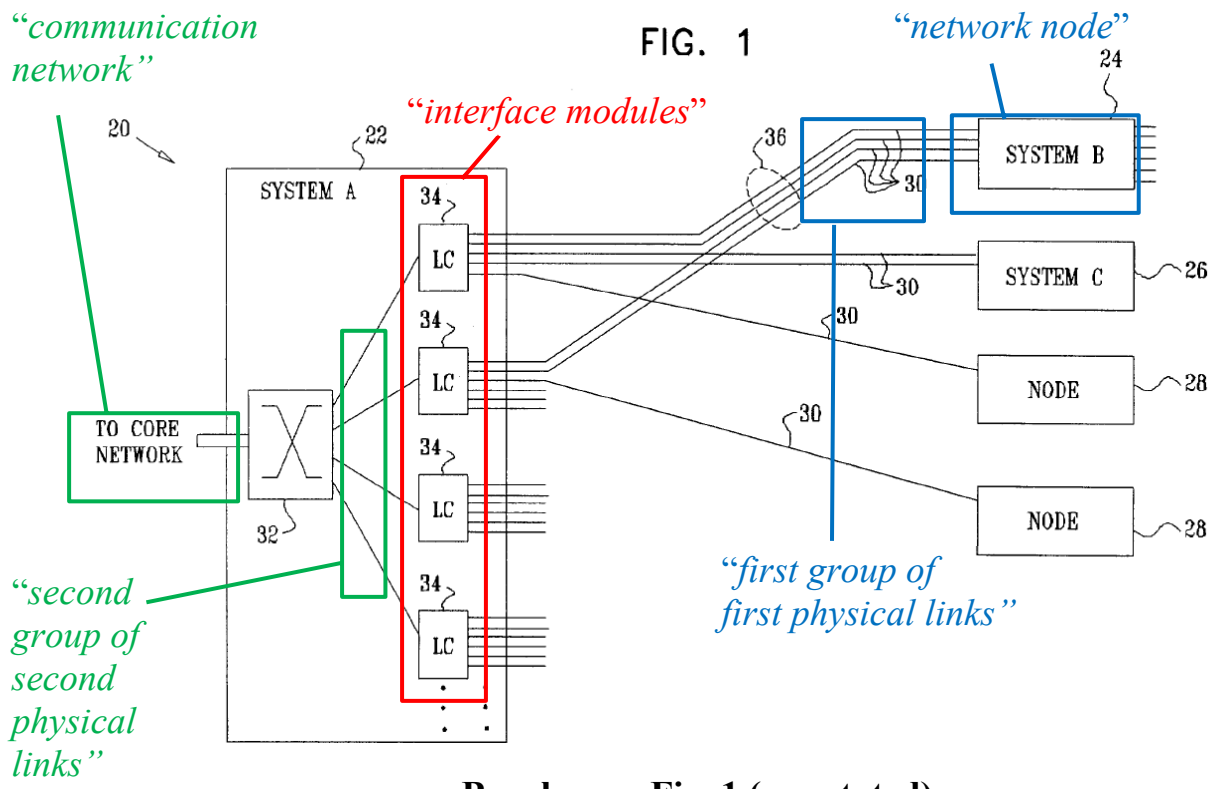


Bruckman, Fig. 2 (annotated).

68. As shown in Fig. 2 above, each of the traces connects between the

switching core 40 and one of the concentrators 50 and are thus arranged in parallel. A POSITA also would have understood that traces on a printed circuit board are *physical* links. Bruckman explains "the **link aggregation** function applies not only to links 30 in group 36 but also to traces 52." Bruckman, [0057]. "Link aggregation is a technique by which a group of **parallel physical links** between two endpoints in a data network can be joined together into a single logical link." Bruckman, [0002].

69. As shown in Fig. 1 below, the main card 32 connects to the core network. Thus, by connecting the lines cards 34 ("*interface modules*") to the switching core 40 of the main card 32, the traces 52 connect the line cards 34 to a "*communication network*" as claimed.



Bruckman, Fig. 1 (annotated).

70. Thus, because Bruckman's line cards are connected to a core network using a set of parallel traces, Bruckman renders obvious "*coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel*" as claimed.

[1.4] *at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;*

71. For the reasons explained above at [1.2], Bruckman's system handles traffic in both directions between the core network and the customer nodes. An aggregator distributes traffic not only to full-duplex links 30 (*see* [1.2]), "but also to traces 52 that connect to multiplexers 50 that serve these links." Ex.1005,

[0057]. Since the aggregator includes both a traffic "distributor" and a traffic "collector," it would have been obvious that traffic flows in both directions over traces 52. Ex.1005, [0058], [0065]

72. Accordingly, at least one of the traces 52 ("*at least one of said second physical links*") is "*a bi-directional link operative to communicate in both an upstream direction and a downstream direction*" as claimed.

[1.5] *receiving a data frame having frame attributes sent between the communication network and the network node;*

73. Bruckman's main card 32 includes an aggregator 54 that receives data frames that are sent between the core network ("*communication network*") and links 30 connected to customer nodes (a "*network node*"). The data frames include various attributes, including frame headers ("*frame attributes*"):

> Aggregator 54 comprises a distributor 58, which is responsible for **distributing data frames arriving from the network among links 30** in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the **frame header**, as described in the Background of the Invention.

Bruckman, [0058].

74. Thus, because Bruckman's main card 32 receives data frames from the core network for transmission to the links 30 which connect to customer nodes, and those data frames have headers, Bruckman renders obvious "*receiving a data frame having frame attributes sent between the communication network and the*

network node" as claimed.

[1.6] *selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and*

75. **First**, Bruckman's aggregator 54 selects which link 30 will carry the data frame to the destination customer node by applying a hashing method on a portion of the information in the header of the data frame:

> [0058] Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, **distributor 58 determines the link over which to send each frame based on information in the frame header**, as described in the Background of the Invention. **Preferably, distributor 58 applies a predetermined hash function to the header information**, wherein the hash function satisfies the following criteria:
> [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
> [0060] The hash function uses all the specified input data from the frame headers.
> [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
> [0062] The hash function generates very different hash values for similar data.
> [0063] For example, distributor 58 may implement the hash function shown below in Table I

Bruckman, [0058]-[0063].

76. Bruckman gives an example of a hash function as shown in Table 1, below.

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

"single computation"

Bruckman, Table 1, [0063] (annotated).

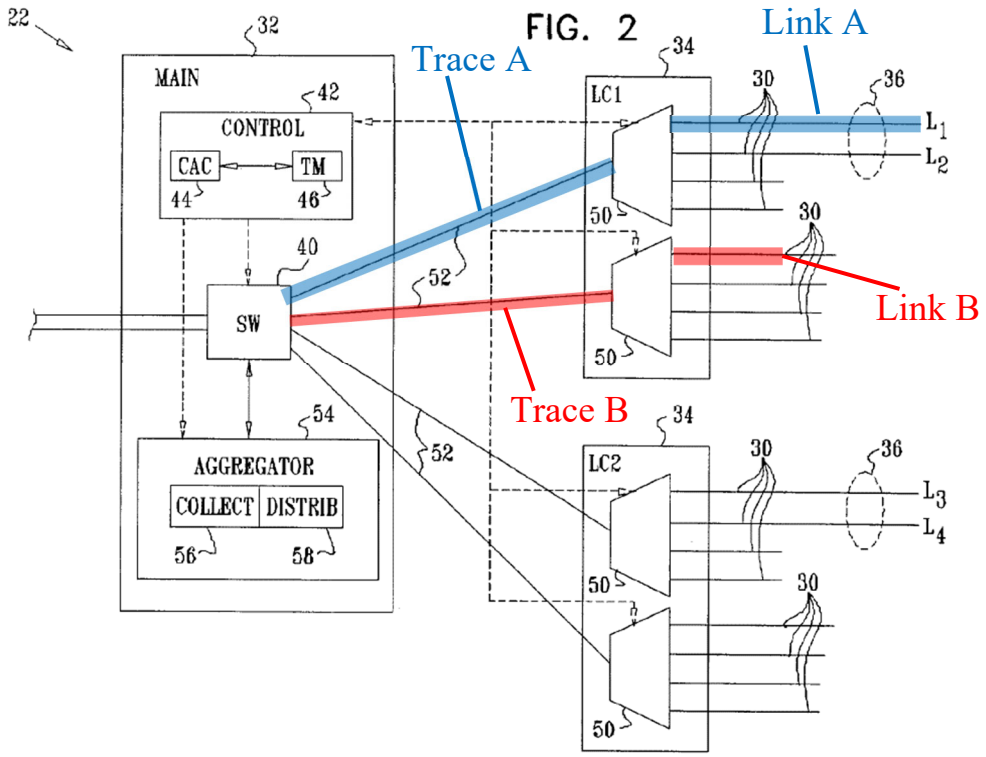
77. Bruckman's hashing function thus selects which link 30 ("*first physical link out of the first group*") will transmit the data frame from the core network to the customer node based on the value calculated from the "hash % lagSize" operation ("*in a single computation*"). A POSITA would have understood that the final line of code in this hashing function, returns the value of the operation "hash % lagSize." A POSITA would have recognized that the code snippet provided in Table 1 is written using the syntax of the C or C++ programming languages. In both C and C++ languages, the % operator refers to a mathematical modulus operation, which returns the remainder from integer division. *See e.g.,* Ex.1020, 7:44-45 ("%: represents the mathematical modulo

operation, i.e., x % y=remainder (x/y)"). Consistent with the discussion above in section VII, either the "hash % lagSize" operation or its result teach the claimed single computation.

78. **Second,** a POSITA would have found it obvious that Bruckman's hashing function selects both which link 30 ("*first physical link out of the first group*") **and** which trace 52 ("*second physical link out of the second group*") will transmit the data frame from the core network to the customer node. Bruckman explicitly states that the hash function determines which link 30 receives the data frame: The "distributor 58, which is responsible for distributing data frames arriving from the network **among links 30** in aggregation group 36 … determines **the link** over which to send each frame based on information in the frame header." Bruckman, [0058].

79. Bruckman provides limited description of how the hash computation is used to select a trace. A POSITA would have recognized, however, that Bruckman's selection of a link 30 also determines which trace 52 will transmit the data frame. For example, referring to Fig. 2 below, if Link A is selected by the hash function, then Trace A is also selected since Trace A is the only trace that connects to link A to the switching core. Similarly, if Link B is selected by the hash function, then Trace B is also selected, since Trace B is the only trace that connects link B to the switching core. Accordingly, selection of a specific link 30

also determines a specific trace 52, e.g., whether to select Trace A or Trace B.



Ex.1005, Fig. 2 (annotated).

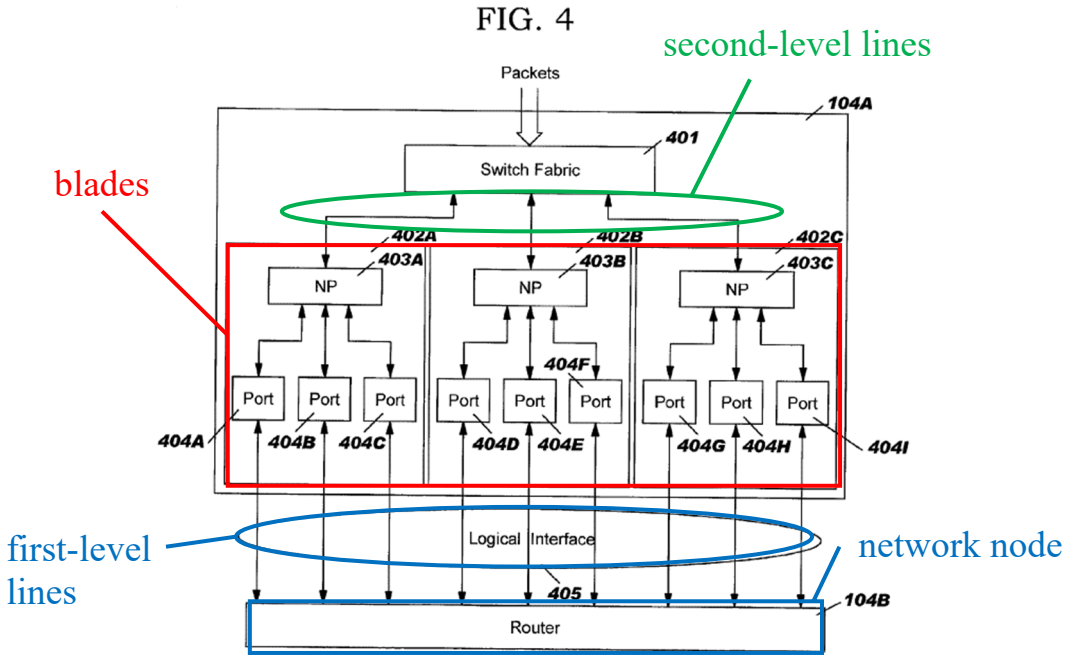
80. A POSITA would have found it obvious that selection of a link 30 also determines which trace 52 will transmit the data frame because each link 30 is associated with only a single trace 52.

81. Alternatively, in accordance with **Ground 2**, it would have been obvious to use the result of a single hash computation to select links from two groups from the combination of Bruckman and Basso. A POSITA would have found it obvious that selection of a link 30 also determines which trace 52 will transmit the data frame because each link 30 is associated with only a single trace

52. *See supra* VIII.A.3. Moreover, using the result of a single hash computation to select links from two groups was known. Basso describes using a single hash computation to select a blade/port combination. Upon receipt of a packet, "a hash function may be performed on the source and destination address in the packet header to generate a hash value." Using that single hash value, Basso's network device selects "an appropriate **blade/port combination** may be identified to transmit the received packet of data." Basso, [0012]. Basso thus uses a single hash computation to select both a first-level line (one of the ports) and second-level line (one of the blades)—a blade/port combination.

82. Basso describes a two-level demultiplexer that is analogous to Bruckman's two-level demultiplexer, as shown in Fig. 4 below.

FIG. 4



Basso, Fig. 4 (annotated).

83. Like Bruckman's links 30, Basso describes a set of physical links between blades of a router 104a to another network node (router 104B in the example of Fig. 4). Like Bruckman's traces 52, each of Basso's blades is connected to the switch fabric, which is connected to a communication network such as the Internet. *See* Basso, Fig. 1. Accordingly, selection of Bruckman's blade/port combination is analogous to selection of a trace 30 and link 52 in Bruckman. For the reasons explained above at VIII.A.3, a POSITA would have found it obvious for Bruckman's hash computation to select a trace and link combination, as evidenced by Basso's hash computation that selects a blade/port combination.

84. Thus because Bruckman describes applying a hash function that selects a link, and a POSITA would have found it obvious that such a hash function also selects a trace, as also evidenced by Basso, Bruckman and Basso render obvious "*selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group*" as claimed.

[1.7] *sending the data frame over the selected first and second physical links,*

85. Bruckman explains that data frames are sent over the selected links:

> Aggregator 54 comprises a distributor 58, which is responsible for **<u>distributing data frames arriving from the network among links 30</u>** in aggregation group 36. Typically, distributor 58 determines the link

> over which to send each frame based on information in the **frame header**, as described in the Background of the Invention.

Bruckman, [0058].

86. Consistent with the example above in [1.6], it would have been obvious that sending a frame over Link A involves sending the data frame over Trace A as well.

87. Thus, because Bruckman describes transmitting frames over a specific link based on the result of the hash function, Bruckman renders obvious "*sending the data frame over the selected first and second physical links*" as claimed.

[1.8] *said sending comprising communicating along at least one of said bi-directional links.*

88. As explained above at [1.2] and [1.4], Bruckman's device transmits data frames in both directions, and thus all of the links 30 and traces 52 are bi-directional. A POSITA would have thus understood that when passing traffic from the core network to the customer node, that when sending traffic across selected links involve sending traffic across bi-directional links. Thus, Bruckman renders obvious "*said sending comprising communicating along at least one of said bi-directional links*" as claimed.

7. Claim 2

[2.1] *The method according to claim 1, wherein the network node comprises a user node, and*

89. Bruckman's customer nodes 24, 26, and 28 are "*user node[s]*" as

claimed. Indeed, Bruckman uses the term customer and user interchangeably. *Compare* Bruckman, [0013] ("**Service level agreements** between network service providers and **customers** commonly specify a certain committed bandwidth") *with* Bruckman, [0072] ("The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the **users' service level agreements**, as is known in the art)").

90. Thus, because Bruckman's nodes 24, 26, and 28 are referred to as customer nodes, and Bruckman uses customer and user interchangeably, Bruckman renders obvious "*wherein the network node comprises a user node*" as claimed.

[2.2] *wherein sending the data frame comprises establishing a communication service between the user node and the communication network.*

91. **First**, as explained above at [1.7], Bruckman's device transmits data frames to customer nodes over the links selected from the link aggregation group.

92. **Second**, Bruckman's disclosure is directed to "**establishing** a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints." Bruckman, abstract; *see also* Bruckman, claims 1, 11, 19, 32. Bruckman explains that establishment of the aggregated links provides a communication service to the customer.

> When aggregated links are used to serve a given customer, the service provider may allocate a certain fraction of the bandwidth on each of the physical links in the aggregation group so that **the aggregated logical link provides the total bandwidth guaranteed by the customer's service level agreement**. Typically, however, the actual bandwidth consumed on each of the physical links fluctuates statistically due to the non-uniform distribution of load among the links in the aggregation group. Furthermore, if one of the physical links fails, the bandwidth consumed on the remaining links in the group will need to increase in order to maintain the minimum guaranteed total bandwidth on the aggregated logical link. Under these circumstances, the service provider may not be able to provide all customers with the minimum bandwidth guaranteed by their service level agreements.
>
> Embodiments of the present invention provide methods **for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees**, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctuation and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.

Bruckman, [0014]-[0015].

93. A POSITA would have understood that the presence of an agreement means that there would obviously have been a service established before the

frames are actually sent and that sending the frames is in furtherance of that established service. Additionally, a POSITA would have found it obvious for transmission of an Ethernet frame to involve establishing a transmission control protocol (TCP) session ("*communication service*") using, for example, a SYN packet. *See, e.g.*, RFC793, 23 (showing that the first action when performing an "active OPEN" of a communication session is to create a transmission control block (TCB) and to send a "SYN" packet), 27 ("For a connection to be established or initialized, the two TCPs must synchronize on each other's initial sequence numbers. This is done in an exchange of connection establishing segments carrying a control bit called 'SYN' (for synchronize) and the initial sequence numbers.").

94. Thus, Bruckman's communication of data frames across the aggregated links provides a service between the communication network and customer nodes, which renders obvious "*wherein sending the data frame comprises establishing a communication service between the user node and the communication network*" as claimed.

8. Claim 3

[3.1] *The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.*

95. **First,** Bruckman's traces 52 ("*second physical links*") are formed on a printed circuit back plane and are thus "*backplane traces*" as claimed.

> Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40. **Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52**. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of statistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admitted on each link 30 so that it remains between a predetermined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the operational status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concentrators 50, based on the status information and the bandwidth allocations made by CAC 44.

Bruckman, [0056].

96. **Second**, because the line cards 34 ("*one or more interface modules*") are arranged in a card rack and plugged into the printed circuit board plane, the line cards are "*coupled*" to the printed circuit back plane.

97. Thus, because the traces 52 are formed on a printed circuit back plane

to which the line cards 34 are plugged into, Bruckman renders obvious "*wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled*" as claimed.

9. Claim 4

[4.0] ***A method for communication, comprising:***

98. See [1.0].

[4.1] ***coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;***

99. See [1.1].

[4.2] ***coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;***

100. See [1.3].

[4.3] ***receiving a data frame having frame attributes sent between the communication network and the network node;***

101. See [1.5]

[4.4] ***selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and***

102. See [1.6]

[4.5] ***sending the data frame over the selected first and second physical links,***

103. See [1.7]

[4.6] ***at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.***

104. First, Bruckman explains that the links 30 ("*first group of physical*

links") may be organized into link aggregation groups.

> Equipment 22 and certain customer nodes, such as nodes 24 and 26, are configured to serve as aggregation systems in accordance with the above-mentioned Clause 43 of the 802.3 standard. (Equipment 22 and nodes 24 and 26 are accordingly labeled as System A, B and C, respectively.) For example**, an aggregation group 36 of four physical links is defined between equipment 22 and node 24**. Another aggregation group of two physical links may be defined between equipment 22 and node 26. Each aggregation group (as well as each non-aggregated link 30) may serve multiple customer connections between the respective customer node and equipment 22

Bruckman, [0048].

> An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a programmable logic component, such as a gate array. In the example shown in FIG. 2, **aggregation group 36 comprises links L1 and L2, which are connected to LC1, and links L3 and L4, which are connected to LC2**. This arrangement is advantageous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation

> function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation function may reside on line card 34.

Bruckman, [0057].

105. Second, Bruckman explains that the link aggregation groups (LAGs) may be Ethernet LAGs.

> **Although the embodiments described herein refer specifically to link aggregation in Ethernet (IEEE 802.3) networks**, the principles of the present invention may similarly be used in other types of link aggregation, such as Inverse Multiplexing over ATM (IMA) and multi-link connections using the Point-to-Point (PPP) protocol.

Bruckman, [0017].

> **For Ethernet networks, link aggregation is defined by Clause 43 of IEEE Standard 802.3**, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (2002 Edition), which is incorporated herein by reference. Clause 43 defines a link aggregation protocol sublayer, which interfaces between the standard Media Access Control (MAC) layer functions of the physical links in a link aggregation group and the MAC clients that transmit and receive traffic over the aggregated links. The link aggregation sublayer comprises a distributor function, which distributes data frames submitted by MAC clients among the physical

> links in the group, and a collector function, which receives frames over the aggregated links and passes them to the appropriate MAC clients.

Bruckman, [0003].

106. Thus, because Bruckman's set of links 30 are Ethernet links that form a link aggregation group, Bruckman renders obvious "*at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group*" as claimed.

10. Claim 5

[5.0] ***A method for communication, comprising:***

107. See [1.0].

[5.1] ***coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;***

108. See [1.1].

[5.2] ***coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;***

109. See [1.3].

[5.3] ***receiving a data frame having frame attributes sent between the communication network and the network node;***

110. See [1.5].

[5.4] ***selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and***

111. See [1.6].

[5.5] ***sending the data frame over the selected first and second physical links,***

112. See [1.7].

[5.6] ***coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group***

113. See [4.6].

[5.7] ***so as to increase a data bandwidth provided to the network node.***

114. Bruckman explains that the purpose of a link aggregation group is to increase bandwidth.

> Link aggregation is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link. Traffic transmitted between the endpoints is distributed among the physical links in a manner that is transparent to the clients that send and receive the traffic. **Link aggregation offers benefits of increased bandwidth**, as well as increased availability, since the logical link can continue to function (possibly with reduced bandwidth) even when one of the physical links fails or is taken out of service

Bruckman, [0002].

115. Thus, because link aggregation is used to increase bandwidth, Bruckman renders obvious using a LAG "*so as to increase a data bandwidth provided to the network node*" as claimed.

11. Claim 6

[6.1] ***The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and***

116. Bruckman's line cards 34 ("*interface modules*") include concentrators and thus the line cards 34 multiplex traffic from the network node to the communication network.

> Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30. **<u>The concentrators multiplex data traffic between links 30 and traces 52</u>**, which connect the concentrators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of statistical multiplexing. To prevent overloading of traces 52, **<u>concentrators 50</u>** may limit the rate of incoming data admitted on each link 30 so that it remains between a predetermined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the operational status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by

> concentrators 50, based on the status information and the bandwidth allocations made by CAC 44.

Bruckman, [0056].

117. Bruckman also uses the term concentrators and multiplexers interchangeably. *See* Bruckman, [0056] (referring to "concentrators 50"); Bruckman, [0057] (referring to "multiplexers 50").

118. Thus, because Bruckman's concentrators/multiplexers 50 multiplex traffic from the customer nodes to the communication network, Bruckman renders obvious "*wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network*" as claimed.

[6.2] *demultiplexing downstream data frames sent from the communication network to the network node.*

119. A POSITA would have understood that demultiplexing is the opposite of multiplexing. *See generally* Holdsworth. As explained above at [1.2] and [1.4], Bruckman's links and traces are bi-directional. "Aggregator 54 comprises **a distributor 58, which is responsible for distributing data frames arriving from the network among links 30** in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header." Bruckman, [0058]. A POSITA would have understood that when the line card 34 receives a signal on a particular trace, and then selects one of

several links with which to transmit the signal, the line card 34 is performing a demultiplexing function. In other words, while traffic in the upstream direction (customer node to communication network) is multiplexed by the line cards 34, downstream traffic (communication network to customer node) is demultiplexed by the line cards 34.

120. Thus, because Bruckman's links are bi-directional and traffic in the downstream direction is demultiplexed by the line cards, Bruckman renders obvious "*demultiplexing downstream data frames sent from the communication network to the network node*."

12. Claim 7

[7.1] ***The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.***

121. Bruckman explains that its hashing technique for selecting a link "distributes traffic in an approximately uniform manner across the entire set of possible hash values." Bruckman, [0061]. By using a hashing function that distributes the load equally among possible hash values (which correspond to specific links), Bruckman's hashing technique is designed to balance data among the links. Moreover, Bruckman incorporates by reference known algorithms for load balancing data traffic among links:

> **Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distribution algorithms that meet the requirements of the standard, while providing some measure of load balancing among the physical links in the aggregation group**. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information associated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conversations to ports could thus include one or more of the following pieces of information:

Bruckman, [0005].

122. Thus, because Bruckman's hashing technique seeks to distribute the load across the links in a uniform manner, Bruckman renders obvious "*wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links*" as claimed.

13. Claim 8

[8.1] ***The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.***

123. As explained above at [1.6], Bruckman's selection process includes applying a hashing function to the header. Bruckman's hashing function is shown in Table 1 below:

hashing function "*mapping function*"

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

Bruckman, Table 1 (annotated).

124. Bruckman's hashing function is a mapping function because it maps a particular data frame to a specific trace 52 and link 30 through the system. Further, claim 9 explicitly recites that a mapping function includes a "*hashing function*."

125. Thus, because Bruckman's selecting process involves a hashing function, Bruckman renders obvious "*wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes*" as claimed.

14. Claim 9

[9.1] *The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.*

126. As explained above at [8.1], Bruckman's hash function is a mapping function.

hash function "*hashing function*"

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

Bruckman, Table 1 (annotated).

127. Thus, because Bruckman's mapping function includes a hashing function, Bruckman renders obvious "*wherein applying the mapping function comprises applying a hashing function*" as claimed.

15. Claim 10

[10.0] ***The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,***

128. As shown in line A, Bruckman's hash function receives two parameters: *hdr[3] and lagSize. Bruckman explains that "lagsize is the number of

[3] A POSITA would have understood that in many programming languages, including C, the asterisk in front of the variable "hdr" indicates that the variable hdr is a pointer that stores the address of the variable. The asterisk serves to

active ports (available links 30) in link aggregation group." Bruckman, [0064].

DISTRIBUTOR HASH FUNCTION

```
A   unsigned short hash(unsigned char *hdr, short lagSize)
    {
    short i;
    unsigned short hash=178; // initialization value
    for (i=0; i<4; i++)          // hdr is 4 bytes length
      hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
    return (hash % lagSize)
    }
```

"determining a hashing size"

Bruckman, Table 1 (annotated).

129. Accordingly, by receiving the lagSize parameter, which is the number of active ports (available links 30) in the link aggregation group, Bruckman's hashing function determines "*a hashing size responsively to a number of at least some of the first and second physical links*." Thus, Bruckman renders this limitation obvious.

[10.2] *applying the hashing function to the at least one of the frame attributes to produce a hashing key,*

130. As also shown in Line A, Bruckman's hashing function receives the "hdr" parameter ("*frame attributes*"). "Here hdr is the header of the frame to be

---

"dereference" the pointer and retrieve the value of the variable (not the memory address) stored in hdr.

distributed." Bruckman, [0064]. Then, in Line B, the hdr value is used to produce a "*hashing key*" represented by the variable "hash."

DISTRIBUTOR HASH FUNCTION

```
A   unsigned short hash(unsigned char *hdr, short lagSize)
      {
      short i;
      unsigned short hash=178; // initialization value
      for (i=0; i<4; i++)          // hdr is 4 bytes length
B       hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
      return (hash % lagSize)
      }
```

variable representing "*frame attributes*"

"*hashing key*"

Bruckman, Table 1 (annotated).

131. As will be explained in more detail below, the "hash" variable represents a *hashing key* because it is used in a modulus operation (%) to produce the final hash result. A POSITA would have been familiar with programming terminology and recognized that in many programming languages, including the code shown in Table 1, the % symbol represents the modulus operation. *See* Eckel, 85.

132. Thus, because Bruckman's hashing function uses the header information to produce a hashing key, Bruckman renders obvious "*applying the hashing function to the at least one of the frame attributes to produce a hashing key*" as claimed.

[10.3] ***calculating a modulo of a division operation of the hashing key by the***

hashing size, and

133. Bruckman's hashing function includes, in Line C, the operation "hash % lagSize." As known to POSITAs, the "%" sign is used to represent the modulus (remainder) operation. Eckel, 85. A POSITA would have thus understood that the operation in Line C calculates the remainder ("*modulo*") that results from dividing the value of "hash" ("*hashing key*") by the value of "lagSize" ("*hashing size*").

DISTRIBUTOR HASH FUNCTION

```
A   unsigned short hash(unsigned char *hdr, short lagSize)
      {
      short i;
      unsigned short hash=178; // initialization value
      for (i=0; i<4; i++)          // hdr is 4 bytes length
B        hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
C     return (hash % lagSize)
      }
```

"*hashing size*"

"*hashing key*"

Bruckman, Table 1.

134. Bruckman's hashing function thus returns a "*modulo of a division operation*." In Bruckman's example where there are 16 different links 30, the value of the lagSize parameter ("*hashing size*") is 16. Accordingly, the modulo of dividing the hashing key by the hashing size (16) is a value between 0 and 15, which can be represented by a four-bit binary number. Each of these 16 different possible values of the modulo corresponds to one of the 16 different possible links 30.

135. Thus, Bruckman's hashing function, which returns "hash % lagSize," renders obvious "*calculating a modulo of a division operation of the hashing key by the hashing size*" as claimed.

[10.4] ***selecting the first and second physical links responsively to the modulo.***

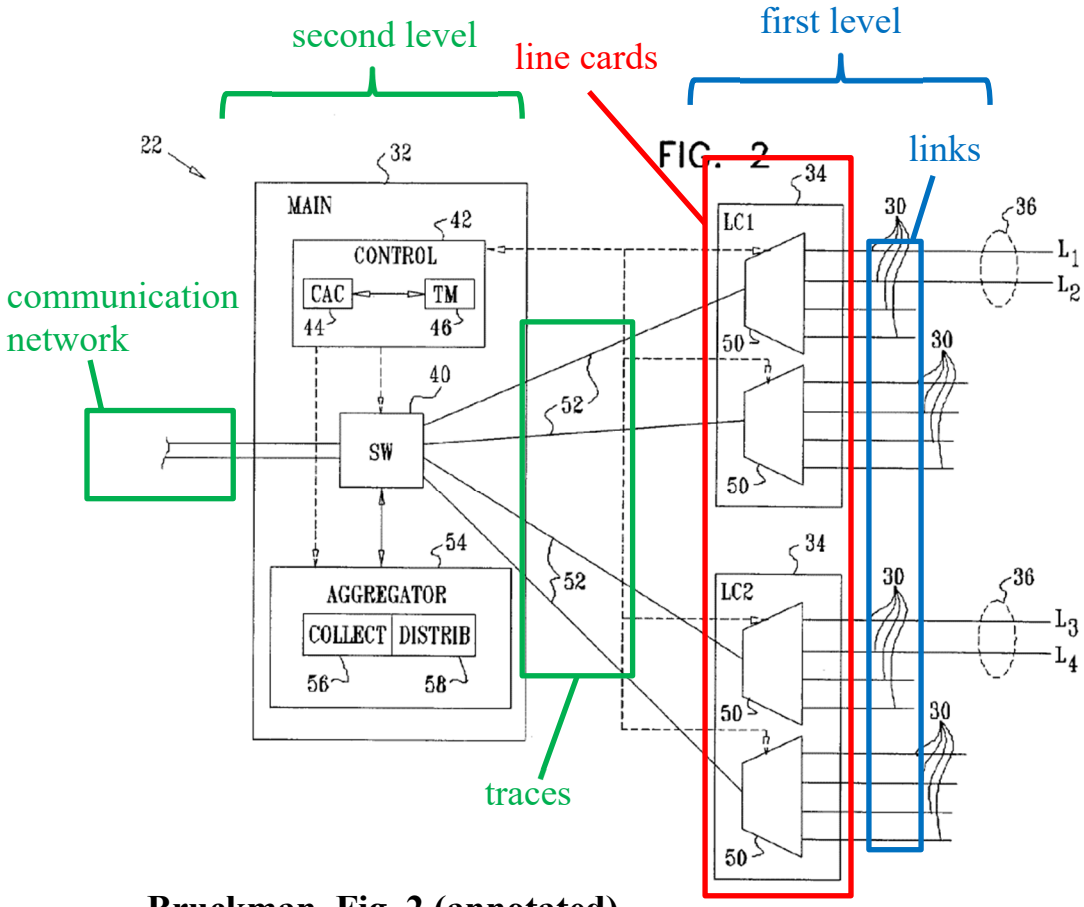
136. For the reasons explained above at [1.6], Bruckman's application of a hashing function teaches selecting first and second physical links responsive to the single hash result. Bruckman's hashing function returns a modulo, as shown in [10.3] by the line "return (hash % lagSize)." The modulus operation or its output (i.e., the modulo) is the single computation that is used to select a trace and a link as described above at [1.6]. Thus, Bruckman renders obvious "*selecting the first and second physical links responsively to the modulo*" as claimed.

16. Claim 11

[11.0] ***The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.***

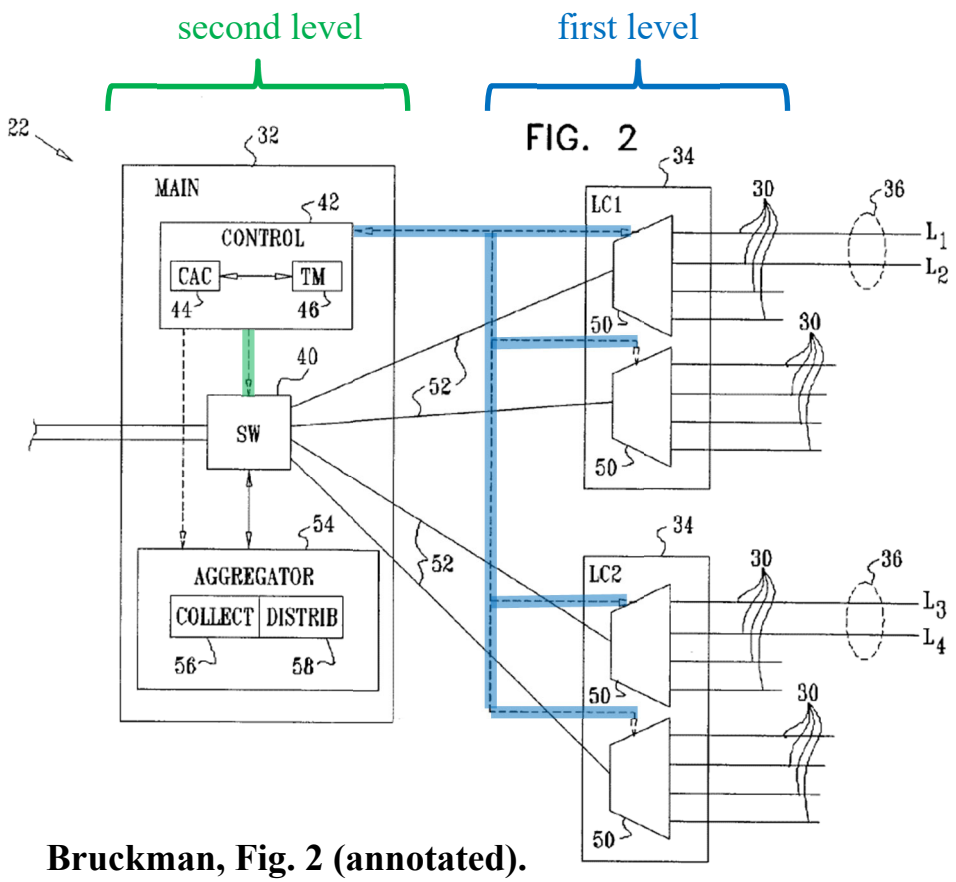
137. This claim limitation merely recites an obvious aspect of digital logic design. As explained above at VIII.A.1, Bruckman's equipment 22 is a two-level structure that resembles a two-level demultiplexer. Bruckman's concentrators 50 act as first-level demultiplexers as each concentrator 50 selects one of four links 30 to transmit data received from a single trace 52. Bruckman's switching core 40 acts as the second-level demultiplexer by distributing data frames from the input

channel amongst four separate traces 52.



Bruckman, Fig. 2 (annotated).

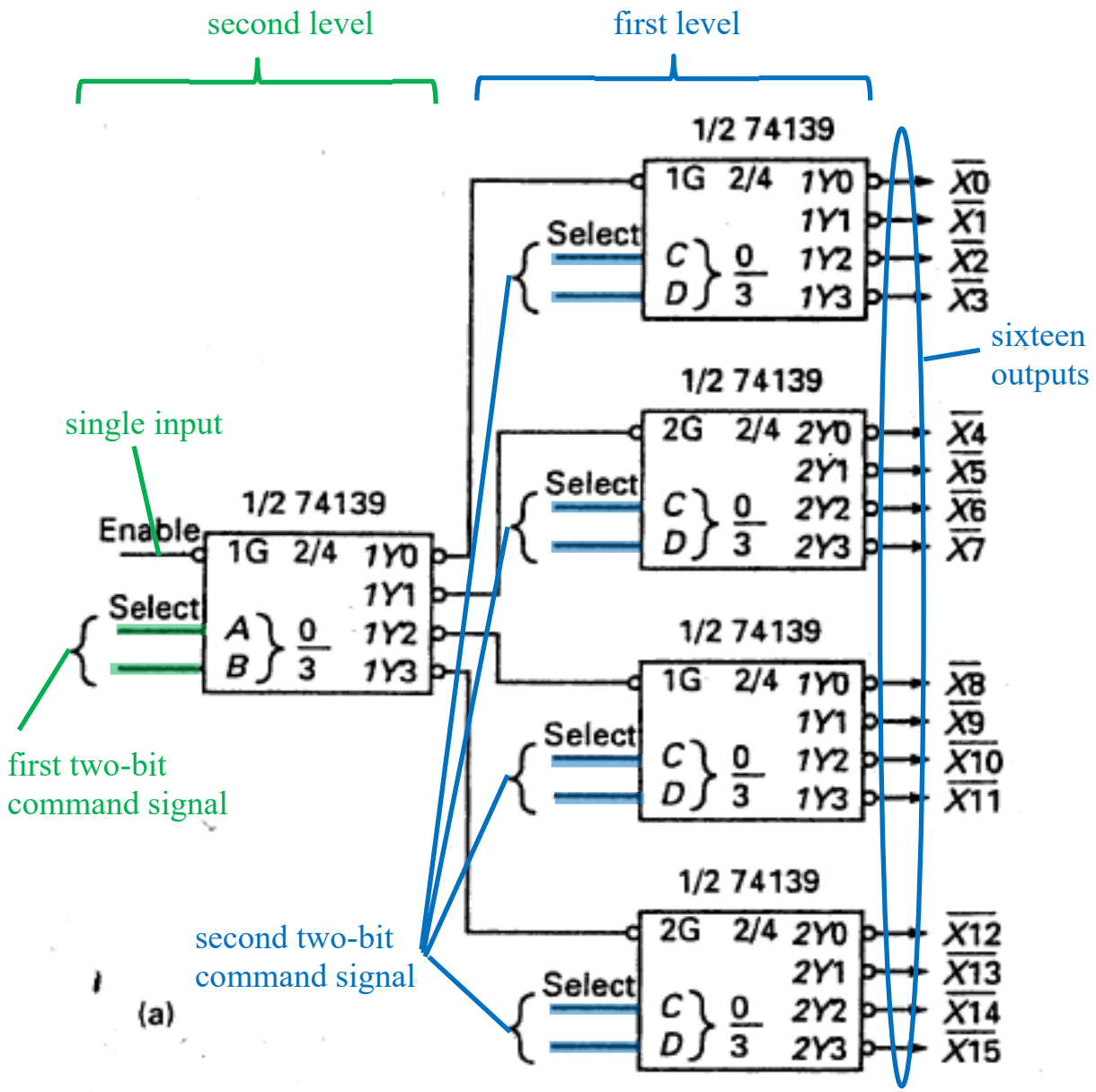
138. Given this well-known, two-level structure, a POSITA would have found it obvious for Bruckman's Equipment 22 to be controlled in a manner consistent with well-known two-level demultiplexer control techniques. Indeed, Bruckman illustrates a control line from the controller 42 to the switching core (second-level demultiplexer) shown below in green. Bruckman also illustrates control lines from the controller 42 to each of the concentrators (first-level demultiplexers) shown below in blue.



Bruckman, Fig. 2 (annotated).

139. Bruckman provides only a limited description of these command lines. Because each of the demultiplexing devices (switching core 40 and concentrators 50) in Fig.2 has one input and four outputs, a POSITA would have found it obvious to control each with a two-bit command signal. A two-bit command signal would efficiently be able to specify one of four operating states, each corresponding to the four possible outputs. A POSITA would have recognized that in Bruckman's example, a two-bit command signal would be used to control the switching core (second-level demultiplexer) and a two-bit signal would be used to control the concentrators (first-level demultiplexers) consistent

with known techniques for controlling two-level demultiplexers. Indeed, as explained in section V.B, such a technique was well-known. *See* Holdsworth, Fig. 5.17(a) (shown below).



Holdsworth, Fig. 5.17(a) (annotated).

140. As explained above at [1.6], Bruckman's hashing function produces a single hash value that is used to select both a line from the first level and a line

from the second level. In Bruckman's example where there are 16 possible links, the lagSize parameter is 16. The operation hash % 16 produces a value ("*modulo*") within the range of 0 and 15. The sixteen different values between 0 and 15 are represented in binary as a four-bit value ("*binary representation of the modulo*").

141. Consistent with known techniques for using a four-bit control signal to control a two-level demultiplexer (formed from 1:4 demultiplexers), two of the four bits are used to control the first level and the remaining two of the four bits are used to control the second level. Thus, a POSITA would have understood Bruckman's control signals (represented by dotted lines in Fig. 2) as two-bit signals. A two-bit signal is applied to the switching core 40 (first-tier demultiplexer) to select a trace 52 and a two-bit signal is applied to all concentrators 50 to select a particular link 30.

142. Thus, because Bruckman illustrates control signals to both levels of a two-level demultiplexer, and a POSITA would have understood the known technique for controlling such a structure involves subsets of bits for control of different levels, Bruckman renders obvious "*selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo*" as claimed.

17. Claim 12

[12.1] ***The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field,***

a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.

143. As explained above at [1.5], Bruckman's equipment 22 receives a data frame with frame attributes. "Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention." Bruckman, [0058]. As noted in Bruckman's Background section:

> The algorithm may make **use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent**. The frame information may be combined with other information associated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conversations to ports could thus include one or more of the following pieces of information:
>
> a) **Source MAC address**
>
> b) **Destination MAC address**
>
> c) Reception port
>
> d) Type of destination address
>
> e) Ethernet Length/Type value
>
> f) Higher layer protocol information

Bruckman, [0005]-[0011].

144. Moreover, POSITAs would have recognized that the list of options recited in claim 12 are well-known components of a layer 2 data frame. *See e.g.*,

RFC 1042.

145. Thus, because Bruckman's equipment receives a data frame having at least a source MAC address and a destination MAC address, Bruckman renders obvious "*wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port*" as claimed.

18. Claim 13

[13.0] ***A method for communication, comprising:***

146. See [1.0]

[13.1] ***coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;***

147. See [1.1].

[13.2] ***coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;***

148. See [1.3].

[13.3] ***receiving a data frame having frame attributes sent between the communication network and the network node;***

149. See [1.5]

[13.4] ***selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and***

150. See [1.6]

[13.5] ***sending the data frame over the selected first and second physical links,***

151. See [1.7]

[13.6] ***coupling the network node to the one or more interface modules and coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and***

152. **First**, as explained above at [1.1] and [1.3], Bruckman renders obvious "*coupling the network node to the one or more interface modules and coupling each of the one or more interface modules to the communication network*."

153. **Second**, Bruckman explains that service level agreements between network service providers (including a "*communication network*") and customers (including a "*customer node*") specify bandwidth in various terms:

> **<u>Service level agreements between network service providers and customers</u>** commonly specify a certain committed bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess bandwidth is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may

> guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. **In general, the bandwidth guaranteed by a service provider, referred to as the peak information rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR)**. The term "guaranteed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.

Bruckman, [0013].

154. Thus, because Bruckman describes that providing guaranteed bandwidth in terms of either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR), Bruckman renders obvious "*coupling the network node to the one or more interface modules and coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node*" as claimed.

[13.7] *allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.*

155. Bruckman describes allocating bandwidth in a manner so as to exceed the guaranteed bandwidth ("*responsively to the bandwidth requirements*"). *See* Ex.1005, [0030]. Bruckman describes "a method for establishing a connection with

a guaranteed bandwidth for transmitting data between first and second endpoints,"

Bruckman, [0027].

> In other words, if the bandwidth guaranteed to a certain customer is B, and the customer is served by an aggregation group of N links, the minimum guaranteed bandwidth that could be allocated on each of the links would be B/N. The safety margins indicate the amount by which the bandwidth allocation must be increased above B/N **in order to fulfill the guaranteed bandwidth requirement of the service level agreement**. Any remaining excess bandwidth on the links in the aggregation group can be used for non-guaranteed, "best-effort" services.

Bruckman, [0016].

> [A]llocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the measure of fluctuation, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection.

Bruckman, [0030].

156. Accordingly, establishing the communication sessions in Bruckman are for the purpose of providing the guaranteed bandwidth and is thus "*responsively to the bandwidth requirements*."

157. Thus, because Bruckman allocates bandwidth to exceed the

guaranteed bandwidth, Bruckman renders obvious "*allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements*" as claimed.

19. Claim 14

[14.0] ***A method for connecting user ports to a communication network, comprising:***

158. As explained above at [1.0], [1.1], and [1.3], Bruckman describes equipment 22 that connects a plurality of links 30 (which correspond to ports) to a communication network. Bruckman's links, which are associated with ports (Bruckman, [0056] "multiple ports that serve respective links 30") are "*user ports*" because they connect to user nodes, as explained in [2.2]. Thus, Bruckman renders obvious a "*method for connecting user ports to a communication network*" as claimed.

[14.1] ***coupling the user ports to one or more user interface modules;***

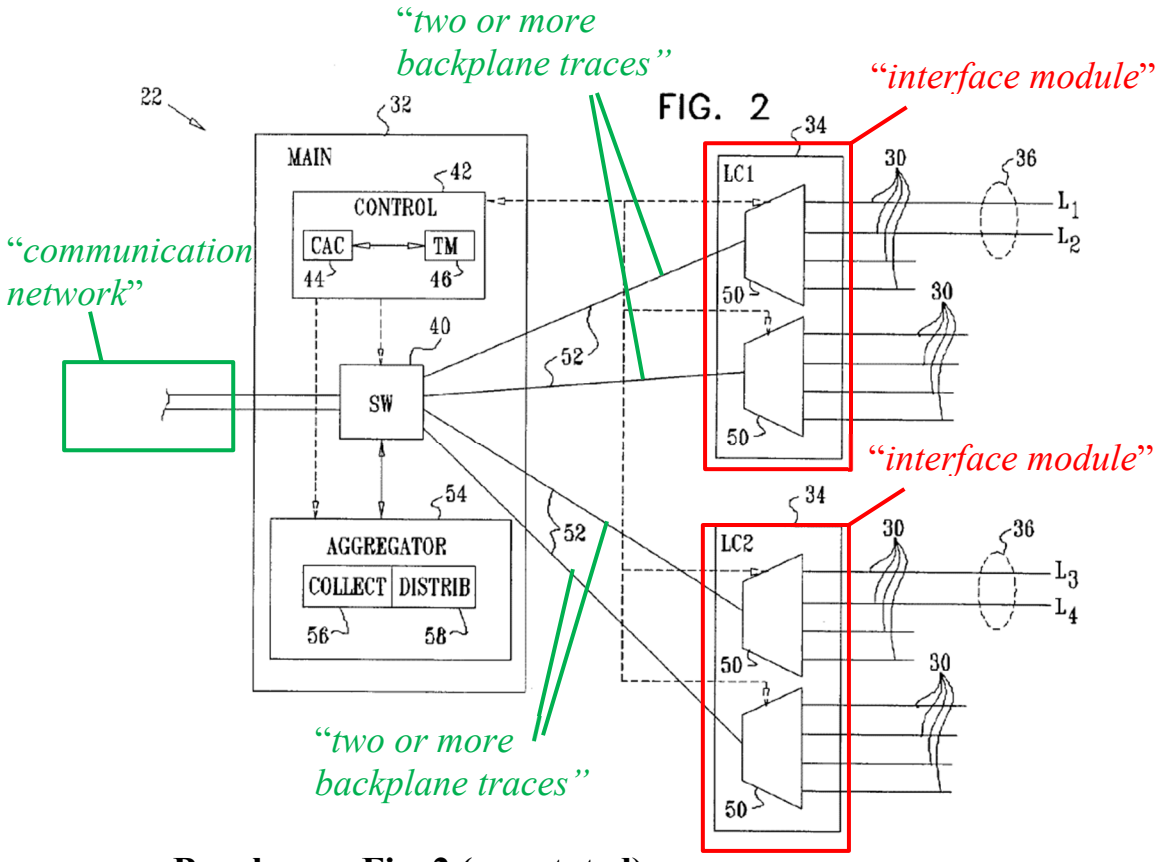
159. Consistent with the discussion at [1.1], Bruckman renders obvious "*coupling the user ports to one or more user interface modules*" as claimed.

[14.2] ***coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,***

160. **First**, consistent with the discussion at [1.3], Bruckman renders obvious "coupling each user interface module to the communication network via a backplane" as claimed. *See also* [3.2] (explaining that Bruckman's traces 52 are

backplane traces).

161. Second, as shown in Fig. 2 below, each of Bruckman's two line cards 34 ("*interface modules*") includes at least two traces 52 that connect that line card to the communication network.



Bruckman, Fig. 2 (annotated).

162. Thus, Bruckman renders obvious "*coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel*" as claimed.

[14.3] ***at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;***

163. Consistent with the discussion above at [1.4], Bruckman renders obvious "*at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction*" as claimed.

[14.4] *receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;*

164. Consistent with the discussion above at [1.6], Bruckman renders obvious "*receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes*" as claimed.

[14.5] *for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and*

165. Consistent with the discussion above at [1.6], Bruckman renders obvious "*for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces*" as claimed. *See also* [3.1] (explaining that Bruckman's traces 52 are backplane traces).

[14.6] *sending the data frame over the selected backplane trace; said sending comprising communicating along said at least one of said backplane traces.*

166. Consistent with the discussion above at [1.7], Bruckman renders obvious "*sending the data frame over the selected backplane trace; said sending comprising communicating along said at least one of said backplane traces*" as claimed. *See also* [3.1] (explaining that Bruckman's traces 52 are backplane

traces).

20. Claim 15

[15.0] ***A method for connecting user ports to a communication network, comprising:***

167. See [14.0]

[15.1] ***coupling the user ports to one or more user interface modules;***

168. See [14.1]

[15.2] ***coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;***

169. See [14.2]

[15.3] ***receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;***

170. See [14.4]

[15.4] ***for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and***

171. See [14.5]

[15.5] ***sending the data frame over the selected backplane trace, at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.***

172. **First**, consistent with the discussion above at [1.7], Bruckman renders obvious "*sending the data frame over the selected backplane trace*" as claimed. *See also* [3.1] (explaining that Bruckman's traces 52 are backplane traces).

173. **Second**, Bruckman explains that the traces 52 are part of a link aggregation group: "As a result of spreading group 36 over two (or more) line

cards, **the link aggregation function applies** not only to links 30 in group 36 but also **to traces 52 that connect to multiplexers 50** that serve these links." Bruckman, [0057]. Also, as explained above at [4.6], Bruckman's link aggregation groups may be Ethernet link aggregation groups.

174. Thus, because Bruckman's equipment 22 sends data over selected traces 52 and links 30, and the traces 52 are part of an Ethernet link aggregation group, Bruckman renders obvious "*sending the data frame over the selected backplane trace, at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group*" as claimed.

21. Claim 16

[16.1] ***The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.***

175. *See* claims 8 and 9.

22. Claim 17

[17.0] ***Apparatus for connecting a network node with a communication network, comprising:***

176. As explained above at [1.1] and [1.3], Bruckman describes Equipment 22 ("*apparatus*") that connects customer nodes (a "*network node*") with a communication network. Thus, Bruckman renders obvious an "*[a]pparatus for connecting a network node with a communication network"* as claimed.

[17.1] ***one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication***

network,

177. First, Bruckman's Equipment 22 includes a plurality of line cards 34 ("*one or more interface modules*").

> FIG. 2 is a block diagram that schematically shows details of equipment 22, in accordance with an embodiment of the present invention. Main card 32 comprises a switching core 40, which switches traffic to and **from line cards 34. Two line cards 34, labeled LC1 and LC2, are shown in the figure**. The operation of switch 40 is managed by a controller 42, typically an embedded microprocessor with suitable software for carrying out the functions described herein.

Bruckman, [0049].

178. Second, Bruckman's line cards process data frames between a customer node ("*network node*") and the communication network.

> **Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40**. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of statistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admitted on each link 30 so that it remains between a

> predetermined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the operational status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concentrators 50, based on the status information and the bandwidth allocations made by CAC 44.

Bruckman, [0056].

179. Third, consistent with the discussion at [1.5], Bruckman's data frames include frame attributes.

180. Thus, because Bruckman's equipment includes a plurality of line cards that multiplex traffic from a customer node to a communication network, and the traffic includes data frames with frame attributes, Bruckman renders obvious "*one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network*" as claimed.

[17.2] *at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;*

181. Consistent with the discussion above at [1.2] and [1.4], Bruckman's Equipment processes traffic in both upstream and down stream directions. Accordingly, the line cards ("*interface modules*") are "*operative to communicate in*

both an upstream direction and a downstream direction." Thus, Bruckman renders obvious "*at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction*" as claimed.

[17.3] *a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules*;

182. Consistent with the discussion above at [1.1], Bruckman renders obvious "*a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules*" as claimed.

[17.4] *a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and*

183. Consistent with the discussion above at [1.3], Bruckman renders obvious "*a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network*" as claimed.

[17.5] *a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;*

184. **First**, Bruckman's Equipment 22 includes a controller 42 ("*control module*"): "The operation of switch 40 is managed by a controller 42, typically an embedded microprocessor with suitable software for carrying out the functions described herein." Bruckman, [0049]. Bruckman further discloses additional

software components running on controller 42, including "Connection Admission Control entity (CAC) 44," "traffic manager 46," and "Aggregator 54." Ex.1005, [0050], [0056]-[0057].

185. Second, consistent with the discussion above at [1.6], Bruckman's controller 42 is "*arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame*" as claimed. Thus, Bruckman renders this limitation obvious.

[17.6] *at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.*

186. *See* [1.2] and [1.4].

23. Claim 18

[18.1] *The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled, wherein the second physical links comprise backplane traces formed on the backplane.*

187. Consistent with the discussion above at [3.1], Bruckman renders obvious "*comprising a backplane to which the one or more interface modules are coupled, wherein the second physical links comprise backplane traces formed on the backplane*" as claimed.

24. Claim 19

[19.0] ***Apparatus for connecting a network node with a communication network, comprising:***

188. *See* [17.0].

[19.1] ***one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;***

189. See [17.1].

[19.2] ***a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;***

190. See [17.3].

[19.3] ***a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and***

191. See [17.4].

[19.4] ***a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,***

192. See [17.5]

[19.5] ***at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.***

193. See [4.6].

25. Claim 20

[20.0] ***Apparatus for connecting a network node with a communication network, comprising:***

194. See [17.0].

[20.1] ***one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;***

195. See [17.1].

[20.2] ***a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;***

196. See [17.3].

[20.3] ***a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and***

197. See [17.4].

[20.4] ***a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,***

198. See [17.5].

[20.5] ***two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.***

199. See [5.6] and [5.7].

26. Claim 21

[21.1] ***The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and***

200. See [6.1].

[21.2] ***demultiplexing downstream data frames sent from the communication***

network to the network node.

201. See [6.2].

27. Claim 22

[22.1] ***The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.***

202. See [7.1].

28. Claim 23

[23.1] ***The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.***

203. See [8.1].

29. Claim 24

[24.0] ***The apparatus according to claim 23, wherein the mapping function comprises a hashing function.***

204. See [9.1].

30. Claim 25

[25.1] ***The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,***

205. See [10.1].

[25.2] ***to apply the hashing function to the at least one of the frame attributes to produce a hashing key,***

206. See [10.2].

[25.3] ***to calculate a modulo of a division operation of the hashing key by the***

hashing size, and

207. See [10.3].

[25.4] ***to select the first and second physical links responsively to the modulo.***

208. See [10.4].

31. Claim 26

[26.0] ***The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.***

209. See [11.1].

32. Claim 27

[27.1] ***The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.***

210. See [12.1].

33. Claim 28

[28.0] ***Apparatus for connecting a network node with a communication network, comprising:***

211. See [17.0]

[28.1] ***one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;***

212. See [17.1].

[28.2] ***a first group of first physical links arranged in parallel so as to couple the***

network node to the one or more interface modules;

213. See [17.3].

[28.3] ***a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and***

214. See [17.4].

[28.4] ***a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,***

215. See [17.5].

[28.5] ***the communication network being arranged to provide a communication service to the network node, the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and***

216. See [13.6].

[28.6] ***the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.***

217. Consistent with the discussion above at [13.7], Bruckman renders obvious "*the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements*" as claimed. A POSITA would have understood that if the links can provide the allocated bandwidth, they are thus "dimensioned" to provide such bandwidth. Otherwise, they would not be able to provide the required

bandwidth.

34. Claim 29

[29.0] ***Apparatus for connecting user ports to a communication network, comprising:***

218. See [17.0].

[29.1] ***one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,***

219. See [17.1].

[29.2] ***at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;***

220. See [17.2].

[29.3] ***a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,***

221. Consistent with the discussion above at [3.1] and [18.1], Bruckman renders obvious "*a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network*" as claimed.

[29.4] ***at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and***

222. Consistent with the discussion above at [1.2] and [1.4], Bruckman renders obvious "*at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction*" as claimed.

[29.5] ***a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.***

223. *See* [19.4].

35. Claim 30

[30.0] ***Apparatus for connecting user ports to a communication network, comprising:***

224. See [17.0] and [14.0].

[30.1] ***one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;***

225. Consistent with the discussion above at [17.1], Bruckman renders obvious "*one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network*."

[30.2] ***a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;***

226. See [29.3].

[30.4] ***a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;***

227. See [19.4].

[30.5] ***at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group***.

228. See [15.5].

36. Claim 31

[31.1] ***The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.***

229. See claims 8 and 9.

B. Grounds 3 & 4: Claims 11 and 26 are obvious under 35 U.S.C. § 103(a) over Bruckman (alone or with Basso) in view of Holdsworth.

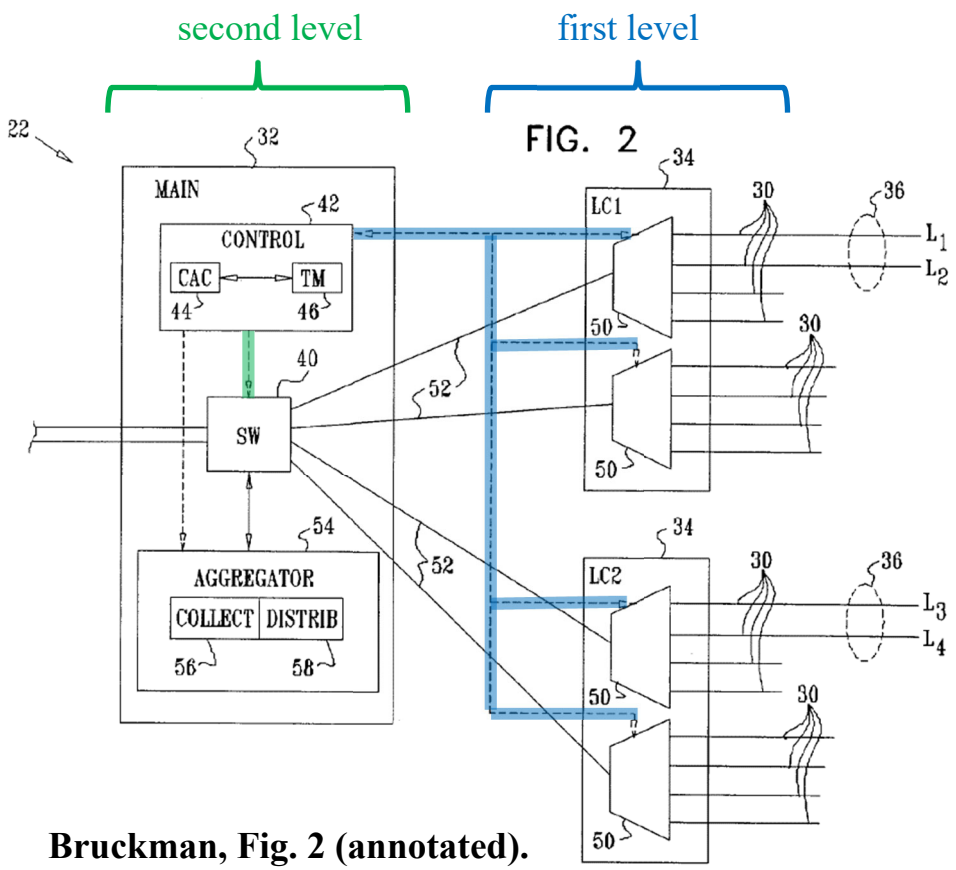
230. As I discussed above regarding Grounds 1 and 2, I believe that various details of implementing Bruckman's two-level demultiplexer design would have been within the background knowledge of a POSITA. To the extent that such information was not background knowledge, it is my opinion that it would have been obvious to a POSITA to refer to a standard textbook for such details. The digital logic design aspects of Bruckman's two-level demultiplexer would have been obvious to a POSITA, as evidenced by Holdsworth.

1. Reasons to Combine Holdsworth with Bruckman and Basso

231. A POSITA would have found it obvious to use basic digital logic

design to implement the control signals for Bruckman's two-level demultiplexer. For example, a POSITA would have found it obvious to use a first subset of bits for a first-level demultiplexer, and a second subset of bits for the second-level demultiplexers, as evidenced by Holdsworth.

232. As explained above, Bruckman's Equipment 22 uses a two-level demultiplexer structure for distributing incoming packets from the communication network among the links 30. Bruckman's depiction of two-level demultiplexer includes dotted lines from the controller to the first-level and second-level demultiplexers, as shown in Fig. 2 below.



Bruckman, Fig. 2 (annotated).

233. A POSITA would have recognized that these dotted lines represent command signals for controlling each of the demultiplexers (switching core 40 and concentrators 50). Bruckman illustrates these control lines but provides few implementation details about the command signals used on the control lines, suggesting that such implementation would have been within the knowledge and skill of a POSITA. Accordingly, a POSITA looking at the teachings of Bruckman would have looked to known implementation techniques for the command signals of Bruckman's two-level demultiplexer. Bruckman explains that "at least some of the functions of the aggregator may be carried out by hard-wired logic or by a programmable logic component, such as a gate array." Bruckman, [0057]. Accordingly, a POSITA would have looked to known logic design techniques.

234. Holdsworth represents basic logic design that is applicable to Bruckman's two-level demultiplexer structure. As stated in the preface, Holdsworth is intended to "act as a reference text for graduates working in this field." Holdsworth, preface. A POSITA would have had a reasonable expectation of success when using Holdsworth's technique because Holdsworth provides an example that is similar to that of Bruckman's example—a plurality of 1:4 demultiplexers that form a 1:16 demultiplexer. A POSITA would have therefore expected that using a two-bit command signal to control a first-level demultiplexer, and a second two-bit command signal to control second-level demultiplexers

would have worked as well in Bruckman's two-level demultiplexer.

235. Given that Bruckman's hash function would produce a hash value in the range of zero to fifteen (and thus be readily represented as a four-bit value), a POSITA would have found it obvious to use two of the four bits to control the second-level demultiplexer (i.e., the switching core) and the other two of the four bits to control the first-level demultiplexers (i.e., the concentrators). Application of Holdsworth technique is also consistent with Basso's technique of using the single hash computation to select both a second-level line (one of the traces 52) and a first-level line (one of the links 30). This would have yielded the predictable result of effectively and efficiently controlling both levels of the two-level demultiplexer using the four-bit hash computation described by Bruckman.

236. Thus, the combination of Holdsworth with Bruckman and Basso represents application of a known technique (Holdsworth's different bit subsets for different levels of a two-level demultiplexer) to a known method (Bruckman's two-level demultiplexer) to yield predictable results (controlling both levels with a single hash computation).

2. Claims 11 and 26

237. The claim analysis for this ground is substantially the same as the analysis provided in ground 1. *See supra* VIII.A.14, VIII.A.29.

IX. CONCLUSION

238. This declaration and my opinions herein are made to the best of my knowledge and understanding, and based on the material available to me, at the time of signing this declaration. I declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 or Title 18 of the United States Code.

Date: December 21, 2022

Henry H. Houh.

Henry Houh